



**Escuela Politécnica Superior**  
**Departamento de Ingeniería Informática**

**INTEGRACIÓN DE LA ARQUITECTURA DE LA INFORMACIÓN  
DENTRO DE UN PROCESO ÁGIL DE DESARROLLO  
CENTRADO EN EL USUARIO**

**TESIS DOCTORAL**

**Luis Alberto Rojas Pino**

**Madrid, Febrero 2017**





**Escuela Politécnica Superior**  
**Departamento de Ingeniería Informática**

**INTEGRACIÓN DE LA ARQUITECTURA DE LA INFORMACIÓN  
DENTRO DE UN PROCESO ÁGIL DE DESARROLLO  
CENTRADO EN EL USUARIO**

**TESIS DOCTORAL**

**Autor:** Luis Alberto Rojas Pino

**Director:** José Antonio Macías Iglesias

**Grupo de Herramientas Interactivas Avanzadas  
(GHIA)**

Memoria presentada para optar al título de Doctor en Ingeniería Informática

**Madrid, Febrero 2017**





**Departamento:** Ingeniería Informática  
Escuela Politécnica Superior  
Universidad Autónoma de Madrid

**Título:** Integración de la Arquitectura de la Información dentro de un proceso ágil de desarrollo centrado en el usuario

**Autor:** Luis Alberto Rojas Pino

**Director:** José Antonio Macías Iglesias

**Miembros del Tribunal de Defensa de la Tesis Doctoral:**

**Presidente:**

**Secretario:**

**Vocal 1:**

**Vocal 2:**

**Vocal 3:**



## Agradecimientos

Estudiar una carrera universitaria fue siempre el sueño de toda mi vida. Recuerdo que ya era inmensamente feliz cuando, después de varios intentos, estaba por fin estudiando la carrera de Ingeniería en Chile. Ahora miro atrás y no puedo creer el mundo de posibilidades que se abrió gracias al esfuerzo y al trabajo constante de todos estos años. Si estudiar ha sido mi sueño, realizar este doctorado ha sido alcanzar y tocar la felicidad sin que existan palabras para describirlo.

La realización de esta tesis doctoral ha sido posible gracias a la ayuda y el apoyo de muchas personas. En primer lugar, quiero agradecer a mi director de la tesis Dr. José Antonio Macías Iglesias por su guía, paciencia y ayuda a través de todos estos años. En momentos difíciles él siempre tuvo la visión para salir adelante y me he sentido afortunado de haber trabajado junto a él.

En el ámbito personal, quiero agradecer especialmente a Milda por su comprensión, paciencia, apoyo y cariño entregado. Ella ha sido mi pilar y refugio de amor y de felicidad todo este tiempo. También agradecer a sus padres, Tatjana y Adomas, quienes me han ayudado sin dudarle y me han tratado siempre como a un integrante más de la familia.

A mi familia, principalmente a mi hijo Gaspar, quien me ha esperado pacientemente y ha tenido siempre palabras de amor y de comprensión. Gracias, hijo, por ser fuerte y esperarme sin pedir nunca nada a cambio. A mis padres, Mario y María, por sólo exigirme estar sano y feliz, y volver pronto. A mis hermanas Elizabeth y Cynthia, por sus constantes muestras de preocupación y de amor.

A mi familia en España, especialmente a Ventura, quien ha sido como una madre para mí, siempre entregándome todo el amor y cuidado. A Mariano, por su amistad y ayuda durante todos estos años. A Mercedes y Rafael, por su ayuda desinteresada. También agradecer a mi amigo Diego, por sus consejos sobre la vida, los negocios y el amor.

Finalmente, agradecer a mi amigo Claudio Gallardo-Araya por las innumerables ocasiones que me ha tendido la mano para ayudarme, y en situaciones muy complejas siempre supo guiarme y aconsejarme.

La realización de la presente tesis doctoral ha sido posible gracias a la subvención del proyecto «Flexor» (TIN2014-52129-R) financiado por el Ministerio de Ciencia e Innovación, y del proyecto «eMadrid-CM» (S2013/ICE-2715) financiado por la Comunidad de Madrid.



## Resumen

El término Arquitectura de la Información (AI) se define como el arte y la disciplina encargada del diseño, análisis, organización, disposición y estructuración de la información. Es un término de relevancia hoy en día que cobra sentido en el desarrollo de aplicaciones y de sistemas de información. De hecho, un aspecto crucial en la AI es que debe promover la usabilidad y, a la vez, ser lo suficientemente flexible para que coincida con las necesidades dinámicas de información de las empresas.

Actualmente, existen diversas propuestas metodológicas para la AI. Sin embargo, la mayoría de las propuestas presentan dificultades para priorizar de manera dinámica los requisitos de usabilidad de contenido, y no cubren completamente su proceso de desarrollo, centrándose sólo en algunas de sus fases. Asimismo, la mayoría de las propuestas requieren de un esfuerzo significativo para adaptarse a entornos complejos, y no proveen una continuidad entre la información de análisis conceptual de la AI y la información de análisis y diseño que necesitan los Analistas e Ingenieros del Software. Todos estos inconvenientes dificultan además la integración de la AI en un entorno de desarrollo ágil, pues aumentan el tiempo para obtener incrementos de software parciales que puedan ser validados por los usuarios finales.

Con el fin de resolver estos inconvenientes, en esta tesis doctoral se aborda la problemática de adaptar las actividades de la AI para el desarrollo de aplicaciones interactivas que mejoren la satisfacción de los usuarios finales, promoviendo una integración de la AI dentro de un marco ágil de desarrollo centrado en el usuario. En concreto, la contribución de esta tesis se compone de las siguientes aportaciones originales:

- Una metodología, llamada Scrum-UIA (**Scrum** driven by **Usable Information Architecture**), cuyo objetivo es integrar la AI en un proceso ágil de desarrollo centrado en el usuario, aportando diferentes actividades, tareas y productos.
- Un método de priorización de requisitos, llamado QMPSR (**Qualitative Method for Prioritizing Software Requirements**), adaptado a la metodología Scrum-UIA, que permite conducir el proceso de priorización a través de las prioridades de la usabilidad, de la AI y del *valor de negocio*.
- Una técnica, llamada InterArch-T (**Interoperable Information Architecture Technique**), para promover el desarrollo incremental en Scrum-UIA de la AI. Dicha técnica se ha implementado mediante la construcción de una herramienta CASE, llamada InterArch (**Interoperable Information Architecture**), que permite al Arquitecto de la Información representar información conceptual de contenidos y transformarla, mediante reglas específicas, en diagramas de clases UML que dan soporte al análisis y diseño de una aplicación interactiva.
- Una herramienta Web, llamada Scrum-UIA-MAT (**SCRUM-UIA MA**nagement **T**ool), que proporciona un sistema de gestión de proyectos basado en Scrum-UIA, y que considera diferentes funcionalidades para apoyar el desarrollo ágil y centrado en el usuario de la AI.

Asimismo, se realizó una verificación de la metodología Scrum-UIA a través de la aplicación de un ejemplo práctico de desarrollo con el apoyo de Scrum-UIA-MAT, centrándose en la implementación de las actividades propuestas, donde los resultados permiten ilustrar la idoneidad de las actividades prescritas para integrar la AI dentro de un modelo de proceso ágil de desarrollo centrado en el usuario. Además, se realizaron distintos experimentos para evaluar QMPSR, InterArch y Scrum-UIA-MAT. De este modo, el método de priorización QMPSR se ha evaluado y comparado con seis conocidos métodos de priorización existentes, con el objetivo de analizar en profundidad las *colisiones* de requisitos software generadas. Los resultados demuestran que QMPSR supera uniformemente a todos los métodos de priorización comparados, generando menos *requisitos colisionados* y presentando un buen comportamiento en términos de *escalabilidad*. Asimismo, se efectuó una evaluación para medir el grado de usabilidad de InterArch con 12 Arquitectos de la Información, obteniendo resultados satisfactorios. De manera similar, se realizó una experimentación con 16 Ingenieros del Software para evaluar la usabilidad de la herramienta Scrum-UIA-MAT, obteniendo también resultados positivos. Las experiencias de los usuarios indican valoraciones positivas y aceptables en cuanto a las dimensiones *Utilidad*, *Facilidad de Uso*, *Facilidad de Aprendizaje* y *Satisfacción*.

**Palabras clave:** Arquitectura de la Información, Diseño Centrado en el Usuario, Desarrollo Ágil, Método de Priorización de Requisitos, Scrum, Usabilidad.

## Abstract

The term Information Architecture (IA) can be defined as the art and discipline responsible for the design, analysis, organization, arrangement and structuring of information. It is a term of relevance today that makes sense in the development of applications and information systems. In fact, a crucial aspect in IA is that it should promote usability while being flexible enough to match the dynamic information needs of the business environments.

Currently, there are various methodological proposals to the IA. However, most of them present difficulties to dynamically prioritize content usability requirements and do not fully cover their development process, focusing only on some of their phases. Likewise, most of the proposals require a significant effort to adapt to complex environments and do not provide a continuity between the information of the conceptual analysis of the IA and the information of analysis and design that Analysts and Software Engineers need. All of these drawbacks also make it difficult to integrate the IA in an agile development environment, as these increase the time to have partial software increments validated by the end users.

In order to solve these problems, this doctoral thesis addresses the problem of adapting the activities of the IA for the development of interactive applications that improve the satisfaction of end users, promoting an integration of the IA into an agile user-centered development framework. Specifically, this thesis is composed of the following original contributions:

- A methodology, called Scrum-UIA (**Scrum** driven by **Usable Information Architecture**), to develop the Information Architecture through an agile user-centered development process, providing different activities, tasks and products.
- A requirement prioritization method, called QMPSR (**Qualitative Method for Prioritizing Software Requirements**), for the Scrum-UIA methodology, which allows to guide the prioritization process through the priorities of usability, IA and *business value*.
- A technique, called InterArch-T (**Interoperable Information Architecture Technique**), to promote incremental development in Scrum-UIA through IA. This technique has been materialized by the construction of a CASE tool, called InterArch (**Interoperable Information Architecture**), which allows the Information Architect to represent conceptual information of contents and to transform it, through specific rules, into diagrams of UML classes that support the analysis and design of an interactive application.

Moreover, a verification of the Scrum-UIA methodology was carried out through an application of a practical example of development with the support of Scrum-UIA-MAT, focusing on the implementation of the proposed activities. The results obtained illustrate the suitability of the activities prescribed to integrate the IA into an agile user-centered process model of development. In addition, different experiments were performed to evaluate QMPSR, InterArch and Scrum-UIA-MAT. Thus, the QMPSR

prioritization method has been evaluated and compared with six well-known prioritization methods, with the objective of analyzing in-depth the *collisions* generated in software requirements rank. The results show that QMPSR outperforms all prioritization methods compared, generating fewer *collision* and presenting no *scalability* problems. An evaluation was also carried out to measure usability in InterArch by enrolling 12 Information Architects, obtaining satisfactory results. Similarly, an experiment was implemented with 16 Software Engineers to evaluate the usability of the Scrum-UIA-MAT tool, also obtaining positive results. User experiences indicate positive and acceptable ratings regarding the dimensions of *Utility*, *Ease of Use*, *Ease of Learning* and *Satisfaction*.

**Keywords:** Information Architecture, User-Centered Design, Agile Development, Prioritization Method, Scrum, Usability.



# Índice de Contenidos

<b>Capítulo 1: Introducción.....</b>	<b>1</b>
1.1. Definición del Problema .....	1
1.2. Hipótesis de Partida y Preguntas de Investigación .....	3
1.3. Metodología de Trabajo.....	5
1.4. Objetivos Principales .....	6
1.5. Solución Propuesta .....	7
1.6. Fortalezas .....	15
1.7. Debilidades .....	16
1.8. Estructura de la Tesis Doctoral.....	18
1.9. Publicaciones .....	19
<b>Capítulo 2: Estado del Arte .....</b>	<b>23</b>
2.1. Mapping Study.....	23
2.1.1. Parámetros .....	23
2.1.2. Ejecución del Estudio .....	31
2.1.3. Resultados Finales .....	35
2.2. Arquitectura de la Información.....	37
2.2.1. Definición .....	37
2.2.2. Arquitecto de la Información .....	38
2.2.3. Entregables de la Arquitectura de la Información.....	38
2.2.4. Metodologías para la Arquitectura de la Información.....	42
2.2.5. Descripción de las Metodologías para la Arquitectura de la Información .....	44
2.2.6. Análisis Comparativo de las Metodologías para la Arquitectura de la Información.....	54
2.3. Metodologías Ágiles .....	59
2.3.1. Definición .....	59
2.3.2. El Manifiesto Ágil.....	59
2.3.3. Descripción de las Metodologías Ágiles.....	61
2.3.4. Viabilidad de las Metodologías Ágiles para Integrar el Diseño Centrado en el Usuario .....	78
2.4. Integración del DCU dentro del Proceso de Desarrollo Ágil.....	86
2.4.1. Prácticas y Artefactos Comunes.....	87
2.4.2. Recomendaciones Específicas para Integrar el DCU en Scrum.....	89
2.5. Priorización de Requisitos .....	93
2.5.1. Definición .....	93
2.5.2. Métodos de Priorización .....	94
2.6. Recapitulación .....	98
<b>Capítulo 3: Una Propuesta para Integrar la Arquitectura de la Información en un Proceso Ágil de Desarrollo Centrado en el Usuario .....</b>	<b>99</b>
3.1. Descripción General de la Propuesta .....	99
3.1.1. Gestión de la Lista del Producto Dirigida por la Investigación Contextual .....	101
3.1.2. Planificación del Sprint Dirigida por la Arquitectura de la Información .....	103
3.1.3. Procesos de Inspección y Mejora Continua Dirigidos por el Usuario Final .....	104
3.2. Proceso Scrum-UIA.....	106
3.2.1. Descripción del Marco de Integración de Scrum-UIA.....	106
3.2.2. Descripción del Modelo de Proceso de Scrum-UIA .....	109
3.2.3. Prescripción de las Actividades Propuestas .....	118
3.2.4. Productos .....	146
3.3. QMPSR.....	181
3.3.1. Descripción Formal de QMPSR .....	181

3.3.2.	Ejemplo de Aplicación.....	189
3.3.3.	Recapitulación .....	200
3.4.	InterArch-T.....	201
3.4.1.	Lineamientos Generales.....	201
3.4.2.	El Diseño Arquitectónico.....	202
3.4.3.	Descripción de la Herramienta CASE InterArch .....	204
3.4.4.	Reglas de Transformación .....	211
3.4.5.	Caso de Uso .....	214
3.5.	Scrum-UIA-MAT .....	219
3.5.1.	Análisis .....	219
3.5.2.	Diseño .....	221
3.5.3.	Descripción de Scrum-UIA-MAT.....	226
3.6.	Recapitulación .....	249
<b>Capítulo 4:</b>	<b>Verificación.....</b>	<b>251</b>
4.1.	Método de Verificación .....	251
4.2.	Aplicación de Gestión de Aprendizaje para el Caso de Estudio.....	251
4.3.	Planificación de Actividades .....	252
4.4.	Ejecución de las Actividades de Verificación .....	256
4.4.1.	Actividad 1.1. Identificación y análisis del contexto .....	256
4.4.2.	Actividad 1.2. Identificación y análisis de contenidos.....	258
4.4.3.	Actividad 1.3. Identificación y análisis de usuarios finales .....	262
4.4.4.	Actividad 1.4. Generación del panorama general del proyecto.....	267
4.4.5.	Actividad 2.1. Identificación y análisis de requisitos del proyecto.....	272
4.4.6.	Actividad 2.2. Generación final de requisitos .....	277
4.4.7.	Actividad 2.3. Determinación de la definición de “Terminado” .....	297
4.4.8.	Actividad 2.4. Determinación del objetivo del Sprint.....	299
4.4.9.	Actividad 3.1. Determinación de la capacidad del Equipo de Desarrollo .....	299
4.4.10.	Actividad 3.2. Refinamiento del objetivo del Sprint.....	303
4.4.11.	Actividad 3.3. Selección de requisitos para el Sprint.....	304
4.4.12.	Actividad 3.4. Generación de tareas de desarrollo.....	306
4.4.13.	Actividad 3.5. Planificación de las tareas de desarrollo.....	313
4.4.14.	Actividad 4.1. Inspección de tareas de desarrollo.....	316
4.4.15.	Actividad 4.2 Revisión del incremento del producto .....	318
4.4.16.	Actividad 4.3 Inspección de prácticas de trabajo .....	319
4.5.	Recapitulación .....	319
<b>Capítulo 5:</b>	<b>Evaluación.....</b>	<b>323</b>
5.1.	Evaluación de las Colisiones de Requisitos en QMPSR .....	323
5.1.1.	Marco Comparativo para la Evaluación Empírica de los Métodos de Priorización .....	324
5.1.2.	Marco Experimental.....	326
5.1.3.	Ejecución de los Experimentos .....	330
5.1.4.	Resultados .....	335
5.1.5.	Discusión .....	336
5.1.6.	Recapitulación .....	346
5.2.	Evaluación de la Usabilidad de InterArch .....	347
5.2.1.	Marco Experimental.....	347
5.2.2.	Ejecución de la Experimentación.....	352
5.2.3.	Resultados Obtenidos.....	353
5.2.4.	Discusión de los Resultados.....	360
5.3.	Evaluación de la Usabilidad de Scrum-UIA-MAT.....	363
5.3.1.	Marco Experimental.....	363

5.3.2. Ejecución de la Experimentación.....	365
5.3.3. Resultados Obtenidos.....	366
5.3.4. Discusión de los Resultados.....	383
5.4. Recapitulación .....	385
<b>Capítulo 6: Conclusiones y Trabajo Futuro .....</b>	<b>387</b>
6.1. Principales Conclusiones de la Tesis Doctoral .....	387
6.2. Lecciones Aprendidas.....	392
6.3. Trabajo Futuro .....	393
<b>Referencias.. .....</b>	<b>397</b>
<b>Anexos.....</b>	<b>411</b>
Anexo A: Cuestionario de Usabilidad .....	411
Anexo B: Descripción de las Tarea Propuestas para Evaluar Scrum-UIA-MAT .....	414

## Índice de Tablas

Tabla 1: Preguntas de investigación.....	5
Tabla 2: Criterios de evaluación para examinar cada documento en las áreas de análisis.....	26
Tabla 3: Ejemplo de aplicación de los criterios de evaluación. ....	27
Tabla 4: Categorías para filtrar documentos importantes. ....	28
Tabla 5: Ejemplos de aplicación de los criterios de evaluación.....	29
Tabla 6: Número de documentos encontrados en las bases de datos. ....	32
Tabla 7: Resultado de la primera evaluación. ....	32
Tabla 8: Documentos seleccionados después de la primera evaluación. ....	33
Tabla 9: Resultado de la segunda evaluación.....	34
Tabla 10: Documentos seleccionados después de la segunda evaluación.....	35
Tabla 11: Documentos seleccionados finalmente y su evaluación en las áreas de análisis. ....	36
Tabla 12: Número de documentos seleccionados finalmente, según el área de análisis y su relevancia. ....	36
Tabla 13: Descripción de las metodologías para la Arquitectura de la Información. ....	53
Tabla 14: Resumen del análisis comparativo de las fuentes de estudios seleccionadas.....	55
Tabla 15: Técnicas recomendadas para integrar la AI centrada en los usuarios y ágil. ....	58
Tabla 16: Resumen de características de las metodologías ágiles. ....	77
Tabla 17: Análisis de la viabilidad de las metodologías ágiles para integrar el Diseño Centrado en el Usuario. ....	83
Tabla 18: Resumen, de acuerdo con la bibliografía revisada, que describe las correspondencias entre las técnicas utilizadas para integrar el DCU en metodologías ágiles y las técnicas de usabilidad utilizadas en la metodología Scrum. ....	88
Tabla 19: Requisitos del SI de gestión de aprendizaje de la universidad.....	192
Tabla 20: Aspectos relevantes del SI de gestión de aprendizaje de la universidad.....	193
Tabla 21: Elementos de los aspectos relevantes del SI de gestión de aprendizaje de la universidad. ....	194
Tabla 22: Priorización de los requisitos, basada en los elementos de los aspectos relevantes del SI. ....	196
Tabla 23: Ranking final de los requisitos del SI de gestión de aprendizaje de la universidad..	197
Tabla 24: Identificación de los perfiles en Scrum-UIA-MAT y el tipo de permiso que éstos tienen en cada módulo.....	229
Tabla 25: Roles de Scrum-UIA, utilizados por los miembros del equipo de trabajo informático. ....	252
Tabla 26: Resumen de las principales características de los métodos de priorización.....	328
Tabla 27: Identificación de los métodos evaluados en cada experimento.....	330

Tabla 28: Identificación del esfuerzo de priorización en cada experimento. ....	335
Tabla 29: Porcentaje de requisitos colisionados generados por los métodos a través de todos los experimentos. ....	344
Tabla 30: Pendientes de las rectas ajustadas para los métodos comparados en todos los experimentos. ....	345
Tabla 31: Estadísticas del tiempo (hh:mm:ss) de ejecución de las tareas en el experimento....	369

## Índice de Figuras

Figura 1: Solución propuesta.....	14
Figura 2: Proceso general de estudio sistemático de la literatura.....	31
Figura 3: Modelo de contenido general.....	39
Figura 4: El plano de un sitio Web para una universidad. ....	39
Figura 5: Maqueta de la página principal de un sitio Web para una universidad.....	40
Figura 6: Hoja de cálculo para inventario de contenidos de un sitio Web para una universidad.....	41
Figura 7: Matriz de metadatos para una empresa de tecnología. ....	42
Figura 8: Resumen gráfico de la evaluación de la viabilidad de las metodologías ágiles para integrar el Diseño Centrado en el Usuario. ....	84
Figura 9: Detalles de Scrum-UIA.....	100
Figura 11: Planificación del Sprint Dirigida por la Arquitectura de la Información.....	104
Figura 12: Procesos de Inspección y Mejora Continua Dirigidos por el Usuario Final.....	105
Figura 13: Actividades de la metodología Scrum-UIA.....	106
Figura 14: Diagrama de actividad con la interdependencia entre las actividades y tareas propuestas.....	111
Figura 15: Flujo de Información Interna de cada una de las tareas de las actividades.....	113
Figura 16: Relación entre los productos del modelo de la metodología Scrum-UIA.....	147
Figura 17: Fases de QMPSR. ....	182
Figura 18: La influencia del Factor de Asociación ( $G$ ) sobre el Nivel de Relevancia del Requisito por Aspecto ( $\lambda$ ).....	186
Figura 19: Priorización de requisitos mediante el método de priorización QMPSR.....	199
Figura 20: Detalle arquitectónico de la técnica InterArch-T para promover el desarrollo incremental de Scrum-UIA y la integración de la AI en entornos ágiles.....	203
Figura 21: Interfaz de usuario de InterArch dividida en tres partes: A, B y C.....	205
Figura 22: Iconos de elementos de contenido. ....	206
Figura 23: Iconos de elementos de enlace.....	206
Figura 24: Manipulación para enlazar elementos de contenido en el entorno de trabajo. ....	207
Figura 25: Manipulación en bloque de los elementos en el diagrama. ....	207

Figura 26: Creación de elementos visuales heredados. ....	207
Figura 27: Paleta de colores de formato de línea. ....	208
Figura 28: Formatos para almacenar el diagrama. ....	209
Figura 29: Tecnologías utilizadas en la implementación de InterArch. ....	210
Figura 30: Módulo de InterArch para las reglas de configuración.....	213
Figura 31: Maqueta de registro de cursos para una universidad. ....	215
Figura 32: Diagrama de clases UML obtenido desde la transformación del modelo de contenido de la Arquitectura de la Información.....	217
Figura 33: Infraestructura tecnológica que soporta Scrum-UIA-MAT. ....	221
Figura 34: Modelo conceptual de la base de datos.....	223
Figura 35: Modelo físico de la base de datos. ....	225
Figura 36: Diagrama de flujo de datos – nivel de contexto.....	226
Figura 37: Plano del sitio Web de Scrum-UIA-MAT. ....	228
Figura 38: Interfaz de Scrum-UIA-MAT, dividida en tres partes: A, B y C.....	230
Figura 39: Módulo para ingresar proyectos (Mod-12). ....	231
Figura 40: Módulo para consultar historias de usuarios (Mod-21). Los colores amarillo, rojo y verde indican el nivel de relevancia que tienen los requisitos en cada aspecto relevante del proyecto, de acuerdo al enfoque del método de priorización QMPSR. ....	232
Figura 41: Módulo para gestionar historias de usuarios (Mod-22). ....	234
Figura 42: Módulo para organizar Lista de Proyecto (Mod-23). ....	235
Figura 43: Módulo para configurar Sprints (Mod-31). ....	237
Figura 44: Módulo para agregar Sprint (Mod-32).....	238
Figura 45: Módulo para consultar tareas de desarrollo (Mod-41).....	239
Figura 46: Módulo para gestionar las tareas de desarrollo (Mod-42). ....	241
Figura 47: Módulo para estimar el esfuerzo diario de las tareas de desarrollo (Mod-51), junto con el gráfico del esfuerzo estimado de las tareas de desarrollo. La línea roja del gráfico indica la planificación de las tareas y la línea azul señala la estimación ideal de las tareas. ....	242
Figura 48: Módulo con las opciones de configuración (Mod-61).....	243
Figura 49: Módulo para gestionar los usuarios finales (Mod-62). ....	244
Figura 50: Módulo para gestionar los aspectos relevantes del proyecto (Mod-63).....	245
Figura 51: Módulo para gestionar los elementos de los aspectos relevantes del proyecto (Mod-64). ....	246
Figura 52: Módulo para establecer las prioridades de los aspectos relevantes del proyecto (Mod-65). ....	247
Figura 53: Módulo para gestionar la capacidad del Equipo de Desarrollo (Mod-66).....	248

Figura 54: Productos creados de acuerdo a las actividades que los generan y la identificación de los módulos que se relacionan.....	255
Figura 55: Gestión de los usuarios finales mediante el uso del módulo “Gestionar Usuarios Finales (Mod-62)” de Scrum-UIA-MAT. ....	263
Figura 56: Ingreso y consulta de los aspectos mediante el uso del módulo “Gestionar Aspectos (Mod-63)”.....	267
Figura 57: Ingreso y consulta de los elementos de los aspectos relevante del SI de aprendizaje, mediante el uso del módulo “Gestionar Elementos (Mod-64)”.....	268
Figura 58: Registro de los requisitos en formato de historias de usuarios a través del módulo “Gestionar Historias de usuarios (Mod-22)”.....	282
Figura 59: Configuración de la priorización de los requisitos por parte del Dueño del Producto a través del módulo “Gestionar Historias de usuarios (Mod-22)”.....	291
Figura 60: Consulta de la Lista del Producto con los requisitos descritos, estimados y priorizados del SI de gestión de aprendizaje, mediante el uso del módulo “Consultar Historias de usuarios (Mod-21)”.....	297
Figura 61: Configuración del Sprint mediante el módulo “Agregar Sprint (Mod-32)”.....	298
Figura 62: Determinación de la capacidad del Equipo de Desarrollo, mediante el módulo “Gestionar la Capacidad del Equipo (Mod-66)”.....	301
Figura 63: Configuración de la Lista de Pendientes del Sprint mediante el módulo “Configurar Sprints (Mod-31)”.....	305
Figura 64: Configuración de las tareas de desarrollo, utilizando el módulo “Gestionar Tareas (Mod-42)”.....	307
Figura 65: Consulta de las tareas de desarrollo definidas para el Sprint inicial del SI de gestión de aprendizaje, utilizando el módulo “Consultar Tareas (Mod-41)”.....	313
Figura 66: Determinación de la planificación de las tareas de desarrollo, mediante el módulo “Estimación del Esfuerzo de las Tareas (Mod-51)”.....	314
Figura 67: Diferentes arreglos para los experimentos en términos de PEL y NPD. ....	326
Figura 68: Proceso de Ejecución de Experimento.....	331
Figura 69: Experimento A. Número de requisitos colisionados (eje y) para todos los métodos evaluados: QMPSR, Orientado al Valor, MoSCoW, Weiger, Definición de Producto y AHP. Valores obtenidos como promedio de 10 ejecuciones para el 25, 50, 75, 100, 125, 150, 175 y 200 requisitos priorizados (eje x), teniendo en cuenta $PEL=aleatorio$ y $NPD=4$ . Las barras de error (SD) se incluyen para mostrar la dispersión del método QMPSR.....	337
Figura 70: Experimento B. Número de requisitos colisionados (eje y) para todos los métodos evaluados: QMPSR, Orientado al Valor, MoSCoW, Weiger y Definición de Producto. Valores obtenidos como promedio de 10 ejecuciones para el 25, 50, 75, 100, 125, 150, 175 y 200 requisitos priorizados (eje x), teniendo en cuenta $PEL=aleatorio$ y $NPD=2$ . Las barras de error (SD) se incluyen para mostrar la dispersión del método QMPSR. ....	338
Figura 71: Experimento C. Número de requisitos colisionados (eje y) para todos los métodos evaluados: QMPSR, MoSCoW, Orientado al Valor y AHP. Valores obtenidos como promedio de 10 ejecuciones para el 25, 50, 75, 100, 125, 150, 175 y 200 requisitos priorizados (eje x),	

teniendo en cuenta $PEL=aleatorio$ y $NPD=8$ . Las barras de error (SD) se incluyen para mostrar la dispersión del método QMPSR. ....	339
Figura 72: Experimento D. Número de requisitos colisionados (eje y) para todos los métodos evaluados: QMPSR, MoSCoW, Orientado al Valor, Weiger y Definición de Producto. Valores obtenidos como promedio de 10 ejecuciones para el 25, 50, 75, 100, 125, 150, 175 y 200 requisitos priorizados (eje x), teniendo en cuenta $PEL=mínimo$ y $NPD=4$ . Las barras de error (SD) se incluyen para mostrar la dispersión del método QMPSR. ....	339
Figura 73: Experimento E. Número de requisitos colisionados (eje y) para todos los métodos evaluados: QMPSR, Orientado al Valor, MoSCoW, Weiger y Definición de Producto. Valores obtenidos como promedio de 10 ejecuciones para el 25, 50, 75, 100, 125, 150, 175 y 200 requisitos priorizados (eje x), teniendo en cuenta $PEL=mínimo$ y $NPD=2$ . Las barras de error (SD) se incluyen para mostrar la dispersión del método QMPSR. ....	340
Figura 74: Experimento F. Número de requisitos colisionados (eje y) para todos los métodos evaluados: QMPSR, Orientado al Valor y MoSCoW. Valores obtenidos como promedio de 10 ejecuciones para el 25, 50, 75, 100, 125, 150, 175 y 200 requisitos priorizados (eje x), teniendo en cuenta $PEL=mínimo$ y $NPD=8$ . Las barras de error (SD) se incluyen para mostrar la dispersión del método QMPSR. ....	341
Figura 75: Experimento G. Número de requisitos colisionados (eje y) para todos los métodos evaluados: QMPSR, Orientado al Valor, MoSCoW, Weiger, Definición de Producto y AHP. Valores obtenidos como promedio de 10 ejecuciones para el 25, 50, 75, 100, 125, 150, 175 y 200 requisitos priorizados (eje x), teniendo en cuenta $PEL=máximo$ y $NPD=4$ . Las barras de error (SD) se incluyen para mostrar la dispersión del método QMPSR.....	342
Figura 76: Experimento H. Número de requisitos colisionados (eje y) para todos los métodos evaluados: QMPSR, Orientado al Valor, MoSCoW, Weiger, Definición de Producto, Kano y AHP. Valores obtenidos como promedio de 10 ejecuciones para el 25, 50, 75, 100, 125, 150, 175 y 200 requisitos priorizados (eje x), teniendo en cuenta $PEL=máximo$ y $NPD=2$ . Las barras de error (SD) se incluyen para mostrar la dispersión del método QMPSR.....	342
Figura 77: Experimento I. Número de requisitos colisionados (eje y) para todos los métodos evaluados: QMPSR, Orientado al Valor, MoSCoW y AHP. Valores obtenidos como promedio de 10 ejecuciones para el 25, 50, 75, 100, 125, 150, 175 y 200 requisitos priorizados (eje x), teniendo en cuenta $PEL=máximo$ y $NPD=8$ . Las barras de error (SD) se incluyen para mostrar la dispersión del método QMPSR. ....	343
Figura 78: Maqueta de registro de anuncios para productos de segunda mano. ....	350
Figura 79: Modelo de contenido de registro de anuncios para productos de segunda mano. .	351
Figura 80: Diagrama de clases UML de registro de anuncios para productos de segunda mano. ....	352
Figura 81: Tiempo (hh:mm:ss) promedio empleado por los usuario en la sesión experimental (eficiencia).....	357
Figura 82: Usabilidad de InterArch. Puntuación media para cada variable que se mide en valores normalizados entre 0 y 100%, incluyendo las barras de error ( $\pm \sigma$ ) y una línea de puntos que representa la media.....	358
Figura 83: Correlación entre las dimensiones <i>Facilidad de uso</i> y <i>Facilidad de Aprendizaje</i> ...	360



Figura 84: Correlación entre las dimensiones <i>Satisfacción</i> y <i>Utilidad</i> . ....	360
Figura 85: Navegadores Web utilizados por los usuarios en el experimento.....	367
Figura 86: Grado de éxito en la finalización de las tareas del experimento por parte de los usuarios (eficacia). ....	368
Figura 87: Tiempo (hh:mm:ss) promedio de duración de las tareas de la experimentación (eficiencia).....	370
Figura 88: Gráficos correspondientes a las ocho preguntas (de 3 a 10) de la dimensión <i>Utilidad</i> , con el número de usuarios (frecuencia) que contestaron a cada pregunta (eje y), frente a la puntuación (de 1 a 7) entregada por los usuarios (eje x).....	373
Figura 89: Gráficos correspondientes a las 11 preguntas (de 11 a 21) de la dimensión <i>Facilidad de Uso</i> , con el número de usuarios (frecuencia) que contestaron a cada pregunta (eje y), frente a la puntuación (de 1 a 7) entregada por los usuarios (eje x). ....	375
Figura 90: Gráficos correspondientes a las cuatro preguntas (de 22 a 25) de la dimensión <i>Facilidad de Aprendizaje</i> , con el número de usuarios (frecuencia) que contestaron a cada pregunta (eje y), frente a la puntuación (de 1 a 7) entregada por los usuarios (eje x).....	377
Figura 91: Gráficos correspondientes a las siete preguntas (de 26 a 32) de la dimensión <i>Satisfacción</i> , con el número de usuarios (frecuencia) que contestaron a cada pregunta (eje y), frente a la puntuación (de 1 a 7) entregada por los usuarios (eje x).....	378
Figura 92: Usabilidad de Scrum-UIA-MAT. Puntuación media para cada dimensión que se mide en valores normalizados entre 0 y 100%, incluyendo las barras de error ( $\pm \sigma$ ) y una línea de puntos que representa la media. ....	380
Figura 93: Correlación entre las dimensiones <i>Satisfacción</i> y <i>Facilidad de Uso</i> . ....	381
Figura 94: Correlación entre las dimensiones <i>Satisfacción</i> y <i>Utilidad</i> . ....	382
Figura 95: Correlación entre las dimensiones <i>Facilidad de Aprendizaje</i> y <i>Utilidad</i> . ....	382

## Índice de Productos

Producto 1: Ejemplo de Informe sobre las características principales del contexto (P1).....	148
Producto 2: Ejemplo de Informe sobre las amenazas y oportunidades del contexto (P2).....	149
Producto 3: Ejemplo de Informe sobre la identificación de los contenidos del proyecto (P3). ....	150
Producto 4: Ejemplo de Modelo de contenido general del proyecto (P4).....	152
Producto 5: Ejemplo de Informe sobre el análisis de las prioridades del contenido (P5). ....	153
Producto 6: Ejemplo de Informe preliminar de los usuarios finales del proyecto y sus características (P6). ....	154
Producto 7: Ejemplo de Informe final sobre las necesidades de los usuarios finales (P7).....	155
Producto 8: Ejemplo de Informe final sobre los elementos y los aspectos relevantes del proyecto (P8).....	156
Producto 9: Ejemplo de Catálogo de criterios de aceptación de los usuarios finales (P9).....	157
Producto 10: Ejemplo de Informe preliminar de requisitos del proyecto (P10).....	158

Producto 11: Ejemplo de Catálogo de requisitos analizados del proyecto (P11).	159
Producto 12: Ejemplo de Catálogo final revisado de requisitos del proyecto (P12).	160
Producto 13: Ejemplo de Lista del Producto con los elementos descritos en formato de Historia de Usuario (P13).	161
Producto 14: Ejemplo de Informe de la estimación de los elementos de la Lista del Producto (P14).	162
Producto 15: Ejemplo de Informe de las prioridades de los elementos de la Lista del Producto (P15).	164
Producto 16: Ejemplo de Informe sobre la definición de “Terminado” para el Sprint (P16).	165
Producto 17: Ejemplo de Informe sobre el objetivo inicial para el Sprint (P17).	166
Producto 18: Ejemplo de Informe sobre la duración del Sprint (P18).	167
Producto 19: Ejemplo de Informe sobre la disponibilidad del Equipo de Desarrollo para el Sprint (P19).	168
Producto 20: Ejemplo de Informe sobre la velocidad del Equipo de Desarrollo (P20).	169
Producto 21: Ejemplo de Informe sobre el objetivo refinado del Sprint (P21).	170
Producto 22: Ejemplo de Lista de Pendientes del Sprint con los requisitos para el Sprint (P22).	171
Producto 23: Ejemplo de Informe de las tareas de desarrollo de los requisitos del Sprint (P23).	172
Producto 24: Ejemplo de Diagrama de Clases UML (P24).	173
Producto 25: Ejemplo de Informe de las técnicas ágiles de la Arquitectura de la Información relacionadas con las tareas de desarrollo (P25).	174
Producto 26: Ejemplo de Informe sobre la estimación de las tareas de desarrollo (P26).	175
Producto 27: Ejemplo de Informe de los responsables de las tareas de desarrollo (P27).	176
Producto 28: Ejemplo de Diagrama de la planificación diaria de las tareas de desarrollo (P28).	177
Producto 29: Ejemplo de Informe de la revisión de las tareas de desarrollo (P29).	178
Producto 30: Ejemplo de Informe sobre la revisión del incremento del producto (P30).	179
Producto 31: Ejemplo de Informe sobre la inspección de las prácticas de trabajo (P31).	180

## Índice de Productos de Verificación

Producto de Verificación 1: Informe sobre las características principales del contexto.	257
Producto de Verificación 2: Informe sobre las amenazas y oportunidades del contexto.	258
Producto de Verificación 3: Informe sobre la identificación de los contenidos del proyecto...	260
Producto de Verificación 4: Informe sobre la identificación de los contenidos del proyecto...	260
Producto de Verificación 5: Informe sobre el análisis de las prioridades del contenido.	262

Producto de Verificación 6: Informe preliminar de los usuarios finales del proyecto y sus características. ....	264
Producto de Verificación 7: Informe final sobre las necesidades de los usuarios finales. ....	266
Producto de Verificación 8: Informe final sobre los elementos y los aspectos relevantes del proyecto. ....	270
Producto de Verificación 9: Catálogo de criterios de aceptación de los usuarios finales. ....	271
Producto de Verificación 10: Informe preliminar de requisitos del proyecto. ....	273
Producto de Verificación 11: Catálogo de requisitos analizados del proyecto. ....	277
Producto de Verificación 12: Catálogo final revisado de requisitos del proyecto. ....	281
Producto de Verificación 13: Lista del Producto con los elementos descritos en formato de Historia de Usuario. ....	287
Producto de Verificación 14: Informe de la estimación de los elementos de la Lista del Producto. ....	290
Producto de Verificación 15: Informe de las prioridades de los elementos de la Lista del Producto. ....	296
Producto de Verificación 16: Informe sobre la definición de “Terminado” para el Sprint. ....	298
Producto de Verificación 17: Informe sobre el objetivo inicial para el Sprint. ....	299
Producto de Verificación 18: Informe sobre la duración del Sprint. ....	300
Producto de Verificación 19: Informe sobre la disponibilidad del Equipo de Desarrollo para el Sprint. ....	302
Producto de Verificación 20: Informe sobre la velocidad del Equipo de Desarrollo. ....	303
Producto de Verificación 21: Informe sobre el objetivo refinado del Sprint. ....	304
Producto de Verificación 22: Lista de Pendientes del Sprint con los requisitos para el Sprint. ....	305
Producto de Verificación 23: Informe de las tareas de desarrollo de los requisitos del Sprint. ....	308
Producto de Verificación 24: Diagrama de Clases UML. ....	308
Producto de Verificación 25: Informe de las técnicas ágiles de la Arquitectura de la Información relacionadas con las tareas de desarrollo. ....	310
Producto de Verificación 26: Informe sobre la estimación de las tareas de desarrollo. ....	311
Producto de Verificación 27: Informe de los responsables de las tareas de desarrollo. ....	313
Producto de Verificación 28: Diagrama de la planificación diaria de las tareas de desarrollo. ....	315
Producto de Verificación 29: Informe de la revisión de las tareas de desarrollo. ....	318
Producto de Verificación 30: Informe sobre la revisión del incremento del producto. ....	319
Producto de Verificación 31: Informe sobre la inspección de las prácticas de trabajo. ....	319



# Capítulo 1: Introducción

---

**E**n el presente capítulo, se plantean los fundamentos principales que dan soporte a la investigación llevada a cabo. En primer lugar, en la sección 1.1, se detalla la problemática abordada. En la sección 1.2, se describen las hipótesis de partida y las preguntas de investigación que conducen la tesis doctoral en todas sus fases, las cuales también son ratificadas a lo largo de la presente investigación. Posteriormente, en la sección 1.3, se presenta la metodología de trabajo. En la sección 1.4, se indican los objetivos principales que persigue esta tesis doctoral. Seguidamente, en la sección 1.5, se propone una solución conforme a la problemática planteada y a las hipótesis descritas. En las secciones 1.6 y 1.7, se presentan las fortalezas y las debilidades de esta tesis doctoral, respectivamente. Finalmente, en las secciones 1.8 y 1.9, se describe la estructura y se presentan las publicaciones a las que ha dado lugar esta tesis doctoral.

## 1.1. Definición del Problema

El término Arquitectura de la Información (AI) se define como el arte y la disciplina encargada del diseño, análisis, organización, disposición y estructuración de la información (Lima-Marques y Carnielli, 2016; Rosenfeld et al., 2015; Rascão, 2015). Aunque este término se ha hecho popular en los últimos años, apareciendo en distintas referencias y trabajos (Wentzel et al., 2016; Gregoriades y Vozniuk, 2015; Morville y Rosenfeld, 2006; Erlin et al., 2008; Leon, 2008), todavía es necesario contar con una conceptualización más adecuada de la AI, con el fin de sistematizar, de forma clara y precisa, la construcción de sistemas software, y poner la información correctamente a disposición de los usuarios que la requieran. En general, dentro de la AI, se pueden identificar tres elementos principales: los usuarios, el contenido o la información propiamente dicha, y la organización encargada de la información. Por lo tanto, se hace necesario un buen manejo y disposición de la información, para que ésta sea accesible por los usuarios finales y esté en consonancia con los objetivos de las organizaciones que proporcionan dicha información.

En este contexto, la simbiosis entre la AI y el Diseño Centrado en el Usuario (DCU) es esencial para poder plantear un análisis y diseño de contenidos acorde con los requisitos del usuario y de la usabilidad, lo que dota de consistencia en sí a la propia concepción de la AI (Abelein y Paech, 2015). De hecho, existen distintos trabajos (Wentzel et al., 2016; Lima et al., 2015; Rocha y Freixo, 2015; Guo y Yan, 2011; Hee et al., 2011) que inciden en la importancia de centrar las actividades de la AI en las necesidades de los usuarios finales. No obstante, y a pesar de su inminente importancia, estos trabajos no identifican ni describen las etapas necesarias en el proceso de despliegue de la AI en los términos descritos, lo que dificulta establecer de manera precisa las exigencias requeridas para considerar el análisis y diseño de contenidos usables desde las fases tempranas del proyecto.

Lo anterior pone de relieve que las propuestas metodológicas que intentan integrar la AI mediante un Diseño Centrado en el Usuario (DCU) (Rosenfeld et al., 2015; Reichenauer, 2005; Picchi, 2011; Toub, 2000) todavía presentan inconvenientes para su integración dentro de un proceso de construcción de software, para que se pueda establecer una relación clara entre los usuarios, la información y la empresa, con el fin de desarrollar aplicaciones software que gestionen y presenten la información de manera efectiva.

Sin embargo, gestionar la información no es una tarea sencilla. Hoy en día, a pesar de que vivimos inmersos en la sociedad de la información, la AI todavía no se gestiona adecuadamente en muchas organizaciones (Fabri y Andrews, 2016; Gregoriades y Vozniuk, 2015; Suárez y Moral, 2015), lo que afecta negativamente a sus costes en términos de errores e ineficiencias, y también en la percepción de los usuarios en cuanto a la calidad de servicio (Lacerda y Lima-Marques, 2014; Martin et al., 2010). Además, las organizaciones operan en entornos que se mueven rápidamente y están en constante evolución, siendo un factor adicional que aumenta la complejidad de la gestión de la información en las organizaciones. Así pues, las organizaciones requieren de una correcta AI para adaptarse eficientemente a estos entornos ágiles (Morris et al., 2016), lo que conlleva a la exigencia de contar con nuevas propuestas metodológicas para su desarrollo (Martin et al., 2010).

Actualmente, existen varias propuestas metodológicas para la AI (Rocio y Cerutti, 2016; Rosenfeld et al., 2015; Zhang et al., 2015; Reichenauer, 2005). Sin embargo, todas ellas se basan en modelos tradicionales del proceso de desarrollo y exigen un esfuerzo considerable para ser adaptadas. Por lo tanto, es difícil llevarlas a cabo en entornos cambiantes, donde a menudo se requiere una respuesta ágil y rápida. Asimismo, la mayoría de las propuestas presentan dificultades para priorizar de manera dinámica los requisitos de usabilidad de contenido, y capturar las principales prioridades que se deben considerar para distinguir el nivel de relevancia de los requisitos (Achimugu et al., 2014). Además, estas propuestas tampoco proveen una continuidad entre la información de análisis conceptual de la AI y la información de análisis y diseño que necesitan los Analistas e Ingenieros del Software. Todos estos inconvenientes dificultan, además, la integración de la AI en un entorno de desarrollo ágil, pues aumentan el tiempo para obtener incrementos de software parciales que puedan ser validados por los usuarios finales. Por lo general, las metodologías ágiles surgen como respuesta a la necesidad de adaptarse rápidamente a los cambios de los entornos. Sin embargo, estas metodologías también presentan dificultades al incorporar el Diseño Centrado en el Usuario en el proceso ágil de desarrollo (Choudhary y Rakesh, 2016; Elbanna y Sarker, 2016; Salah et al., 2014), siendo éste un factor primordial para integrar eficientemente la AI. Varios estudios han abordado la necesidad de involucrar a los usuarios finales en el desarrollo ágil (Brhel et al., 2015; Losada et al., 2013; Isomursu et al., 2012; Kuusinen et al., 2012; Memmel et al., 2007), aportando recomendaciones específicas, pero sin proporcionar una visión integral o completa. Diversos autores (Brhel et al., 2015; Cajander et al., 2013) han reportado estos inconvenientes como un incentivo para crear nuevas metodologías o enfoques para

integrar las actividades de la AI dentro de un proceso ágil de desarrollo centrado en el usuario.

## 1.2. Hipótesis de Partida y Preguntas de Investigación

La presente tesis doctoral se apoya en las siguientes hipótesis de partida:

- **H1:** No existe una integración de la Arquitectura de la Información dentro de un proceso ágil de desarrollo centrado en el usuario.
  - **H1.1:** No existen trabajos anteriores que realicen una integración de la Arquitectura de la Información dentro de un proceso ágil de desarrollo centrado en el usuario.
  - **H1.2:** Las propuestas para integrar la Arquitectura de la Información actuales carecen de un modelo de proceso de desarrollo capaz de adaptarse a entornos ágiles y que considere a los usuarios finales durante todo el ciclo de vida del desarrollo de software.
  - **H1.3:** La mayoría de las metodologías ágiles presentan dificultades para integrar el Diseño Centrado en el Usuario y sólo consideran a los usuarios finales de manera parcial durante el desarrollo de software.
  - **H1.4:** Las metodologías ágiles requieren de prácticas y recomendaciones específicas para integrar de manera eficiente el Diseño Centrado en el Usuario.
  - **H1.5:** Los métodos de priorización de requisitos actuales no permiten conducir de manera ágil el proceso de priorización y capturar formalmente las preferencias de la usabilidad, de la Arquitectura de la Información y del *valor de negocio* que definen la prioridad de los requisitos.
- **H2:** Es posible integrar la Arquitectura de la Información dentro de un enfoque ágil de desarrollo centrado en el usuario.
  - **H2.1:** Es posible idear una metodología para integrar la Arquitectura de la Información en un proceso ágil de desarrollo centrado en el usuario.
  - **H2.2:** Es posible idear un método de priorización que permita a la metodología anteriormente ideada (hipótesis de partida **H2.1**) conducir las prioridades de los requisitos a través de las preferencias de la usabilidad, de la Arquitectura de la Información y del *valor de negocio*, generando una clasificación final que permita reducir las *colisiones de requisitos* y presentar un buen comportamiento en términos de *escalabilidad*.
  - **H2.3:** Es posible promover el desarrollo incremental en la metodología anteriormente ideada (hipótesis de partida **H2.1**) a partir de las descripciones conceptuales que el Arquitecto de la Información crea, dentro del dominio del problema, y que evoluciona a fases más cercanas al dominio de la solución. Asimismo, es posible sistematizar esta técnica mediante la construcción de una herramienta CASE (Computer Aided

Software Engineering), fácil de usar y de aprender por el Arquitecto de la Información.

- **H2.4:** Es posible crear una herramienta Web, que proporcione un sistema de gestión de proyectos basado en la metodología anteriormente ideada (hipótesis de partida **H2.1**), mediante un proceso ingenieril centrado en el usuario. De este modo, la herramienta Web desarrollada será fácil de usar y de aprender por parte de sus usuarios potenciales.

A continuación, la Tabla 1 presenta las preguntas de investigación, agrupadas de acuerdo a las hipótesis de partida que se intentan corroborar.

Preguntas de Investigación	H.P.
<b>PI<sub>1</sub>:</b> ¿Qué propuestas existen para integrar la Arquitectura de la Información dentro de un modelo de proceso ágil de desarrollo centrado en el usuario?	<b>H1.1</b>
<b>PI<sub>2</sub>:</b> ¿Qué elementos de la Arquitectura de la Información cubren las propuestas metodológicas existentes?	<b>H1.2</b>
<b>PI<sub>3</sub>:</b> ¿Qué propuestas existentes para la Arquitectura de la Información tienen una descripción completa y detallada de cada uno de los componentes presentados?	
<b>PI<sub>4</sub>:</b> ¿En qué fases del ciclo de vida de la Arquitectura de la Información se centran las propuestas existentes?	
<b>PI<sub>5</sub>:</b> ¿Qué propuestas existentes de la Arquitectura de la Información permiten diseñar sus productos centrados en las necesidades de los usuarios finales durante las diferentes etapas de desarrollo?	
<b>PI<sub>6</sub>:</b> ¿Qué técnicas para integrar la Arquitectura de la Información permiten considerar las necesidades y capacidades de los usuarios finales, y adaptarse a entornos ágiles?	
<b>PI<sub>7</sub>:</b> ¿Qué metodologías de desarrollo de software son comúnmente consideradas como ágiles?	<b>H1.3</b>
<b>PI<sub>8</sub>:</b> ¿Qué enfoque de desarrollo y modelo de proceso son los más habituales en las metodologías ágiles?	
<b>PI<sub>9</sub>:</b> ¿Qué metodología ágil permite una mejor integración del Diseño Centrado en el Usuario?	
<b>PI<sub>10</sub>:</b> ¿Qué técnicas se utilizan frecuentemente en la integración del Diseño Centrado en el Usuario dentro del paradigma ágil, y facilitan integrar la Arquitectura de la Información?	<b>H1.4</b>
<b>PI<sub>11</sub>:</b> ¿Qué prácticas y recomendaciones específicas se utilizan para integrar el Diseño Centrado en el Usuario en la metodología Scrum?	
<b>PI<sub>12</sub>:</b> ¿Cuál es la importancia de tratar las <i>colisiones de requisitos</i> en los entornos ágiles?	<b>H1.5</b>



<b>PI<sub>13</sub></b> : ¿Qué limitaciones de los métodos de priorización actuales dificultan integrar la Arquitectura de la Información dentro de un marco ágil de desarrollo centrado en el usuario?	
<b>PI<sub>14</sub></b> : ¿Permiten los métodos de priorización actuales adaptarse a entornos ágiles y formalizar la evaluación de la priorización de los requisitos mediante el uso de elementos cualitativos relacionados con las prioridades del proyecto?	
<b>PI<sub>15</sub></b> : ¿Es posible reducir las <i>colisiones de requisitos</i> a través del método de priorización propuesto (QMPSR) dirigido por elementos cualitativos?	
<b>PI<sub>16</sub></b> : ¿En qué medida QMPSR supera a otros métodos de priorización en términos de <i>colisiones de requisitos</i> y de <i>esfuerzo de priorización</i> ?	<b>H2.2</b>
<b>PI<sub>17</sub></b> : ¿Cómo se comporta el método QMPSR en términos de <i>escalabilidad</i> , en función del número de colisiones de requisitos observadas, cuando el conjunto de requisitos aumenta en tamaño?	
<b>PI<sub>18</sub></b> : ¿Cuál es la eficacia alcanzada por los usuarios en la realización de un <i>modelo de contenido</i> utilizando InterArch?	
<b>PI<sub>19</sub></b> : ¿Cuál es la eficiencia alcanzada por los usuarios en la realización de un <i>modelo de contenido</i> utilizando InterArch?	<b>H2.3</b>
<b>PI<sub>20</sub></b> : ¿Cuál es la percepción de usabilidad de los usuarios con respecto a la herramienta InterArch?	
<b>PI<sub>21</sub></b> : ¿Cuál es la eficacia alcanzada por los usuarios en la realización de tareas de gestión de un proyecto utilizando Scrum-UIA-MAT?	
<b>PI<sub>22</sub></b> : ¿Cuál es la eficiencia alcanzada por los usuarios en la realización de tareas de gestión de un proyecto utilizando Scrum-UIA-MAT?	<b>H2.4</b>
<b>PI<sub>23</sub></b> : ¿Cuál es la percepción de usabilidad de los usuarios con respecto a Scrum-UIA-MAT?	

**Tabla 1:** Preguntas de investigación.

### 1.3. Metodología de Trabajo

La metodología de trabajo de esta tesis está basada en el método científico. Para ello, y tras introducir el problema a abordar, la tesis irá corroborando cada una de las hipótesis de partida planteadas, contestando a cada una de las preguntas de investigación relacionadas con cada hipótesis y sus correspondientes sub-hipótesis.

En cuanto a la hipótesis **H1**, se presenta un estudio concienzudo del estado del arte para revisar los trabajos relacionados con la investigación. De este modo, para corroborar la hipótesis de partida **H1.1**, en la sección 2.1 se efectúa un estudio sistemático de la literatura, con el fin de buscar contribuciones científicas que realicen una integración de la AI dentro de un proceso ágil de desarrollo centrado en el usuario. Continuando con el estudio, se profundiza en el análisis de las propuestas identificadas para la AI, con el fin de estudiar la capacidad de sus modelos de proceso para realizar un Diseño Centrado en

el Usuario y adaptarse de manera eficiente a ambientes dinámicos, corroborando así la hipótesis de partida **H1.2**. Adicionalmente, con el objetivo de integrar eficientemente la AI, se analizan las características de las metodologías ágiles, con el fin de distinguir su viabilidad para incorporar el Diseño Centrado en el Usuario, corroborando la hipótesis de partida **H1.3**. Similarmente, se examinan diferentes recomendaciones que facilitan la integración del Diseño Centrado en el Usuario dentro del proceso ágil de desarrollo, lo que permite corroborar la hipótesis de partida **H1.4**. Al final de este estudio, y para corroborar la hipótesis de partida **H1.5**, se investigan las características de los métodos de priorización de requisitos, con el objetivo de identificar si estos permiten conducir, de manera ágil, el proceso de priorización a través de las preferencias de la AI, de la usabilidad y del *valor de negocio*.

Respecto a la hipótesis **H2**, en el Capítulo 3 se presenta la solución propuesta. De esta manera, se detalla la descripción del modelo de proceso ágil de desarrollo de la metodología propuesta y la prescripción de sus actividades y tareas, lo que permite corroborar la hipótesis de partida **H2.1**. De igual manera, para corroborar la hipótesis de partida **H2.2**, se describe el método de priorización de requisitos que permite centrar la discusión en los *aspectos y elementos* relacionados con la usabilidad, la AI y el *valor de negocio*. Asimismo, para corroborar la hipótesis de partida **H2.3**, se detalla la técnica que permite implementar las tareas relacionadas con la AI y promover el desarrollo incremental en la metodología ágil propuesta, a partir de la representación inicial conceptual de los contenidos por parte del Arquitecto de la Información. Al final, para corroborar la hipótesis de partida **H2.4**, se presenta la herramienta Web que permite sistematizar la aplicación de la metodología ágil que se propone.

Seguidamente, para corroborar definitivamente la hipótesis de partida **H2.1**, se realiza un experimento para verificar las actividades prescritas en la metodología ágil, junto con el soporte de su herramienta, a través de la aplicación de un ejemplo práctico.

Finalmente, se realizan diferentes experimentos para evaluar los componentes creados para integrar la AI dentro de un marco ágil de desarrollo centrado en el usuario. De este modo, se estudia la capacidad del método de priorización propuesto en la metodología ágil para adaptarse de manera flexible en los entornos dinámicos, y conducir eficientemente el proceso de priorización, permitiendo corroborar la hipótesis de partida **H2.2**. Posteriormente, se detalla la evaluación llevada a cabo con 12 Arquitectos de la Información, para medir el grado de usabilidad alcanzado por la herramienta CASE, que sistematiza la técnica que promueve el desarrollo incremental de la metodología ágil propuesta, corroborando la hipótesis de partida **H2.3**. Por último, se introduce la experimentación llevada a cabo con 16 Ingenieros del Software para medir el grado de usabilidad de la herramienta Web. Esto lleva a la corroboración de la última hipótesis de partida **H2.4**.

## 1.4. Objetivos Principales

El objetivo principal de esta tesis es abordar la problemática mencionada en la definición del problema (sección 1.1) y proveer una integración de la AI dentro de un

proceso ágil de desarrollo centrado en el usuario. De acuerdo a este objetivo principal, se plantean los siguientes objetivos específicos:

- Estudiar el estado de la literatura en busca de aportes concretos relacionados con un enfoque ágil centrado en el usuario para integrar la AI.
- Analizar los modelos de proceso ágiles más comunes, y determinar su compatibilidad para integrar la AI, considerando a los usuarios finales durante todas las etapas.
- Proponer una metodología ágil y centrada en los usuarios finales de base para crear y testear la AI.
- Proporcionar facilidades a los tomadores de decisiones de la metodología propuesta para capturar formalmente las preferencias de la usabilidad, de la AI y del *valor de negocio*, elicitando los requisitos más apropiados en cada caso .
- Generar información de diseño para los Analistas e Ingenieros del Software a partir de las representaciones conceptuales de los contenidos creados por el Arquitecto de la Información, que permita fomentar el desarrollo incremental de la metodología propuesta.
- Validar la generación de información de análisis y diseño mediante la construcción de una herramienta CASE, que satisfaga las necesidades relacionadas con la utilidad, satisfacción, facilidad de uso y de aprendizaje.
- Construir una herramienta Web fácil de usar, para utilizarla en la metodología propuesta como soporte al proceso de desarrollo de software centrado en el usuario, con el fin de permitir la creación de elementos del proyecto de forma sistemática.
- Minimizar el coste, tiempo y esfuerzo, así como aumentar la automatización e interoperabilidad en el desarrollo de aplicaciones Web interactivas.

### 1.5. Solución Propuesta

De acuerdo a la definición del problema, expuesta en la sección 1.1, y a las hipótesis de partida, enumeradas en la sección 1.2, se propone una integración de la AI dentro de un proceso ágil de desarrollo centrado en el usuario, que permite ajustar las actividades de la AI para el desarrollo de aplicaciones interactivas, con el fin de mejorar la satisfacción de los usuarios finales.

En concreto, la solución propuesta está compuesta por:

1. Una metodología, llamada Scrum-UIA (**Scrum** driven by Usable **I**nformation **A**rchitecture), cuyo objetivo es integrar la AI en un proceso ágil de desarrollo centrado en el usuario, aportando diferentes actividades, tareas y productos. Dicha metodología se basa en Scrum y en una combinación de las recomendaciones y prácticas específicas para integrar el DCU en el paradigma ágil.
2. Un método de priorización de requisitos, llamado QMPSR (**Q**ualitative **M**ethod for **P**rioritizing Software **R**equirements), adaptado a la metodología Scrum-UIA,

que permite conducir el proceso de priorización a través de las prioridades de la usabilidad, de la AI y del *valor de negocio*.

3. Una técnica, llamada InterArch-T (**Inter**operable Information **Arch**itecture Technique), para promover el desarrollo incremental en Scrum-UIA de la AI. Dicha técnica se ha implementado mediante la construcción de una herramienta CASE, llamada InterArch (**Inter**operable Information **Arch**itecture), que permite al Arquitecto de la Información representar información conceptual de contenidos y transformarla, mediante reglas específicas, en diagramas de clases UML, que dan soporte al análisis y diseño de una aplicación interactiva.
4. Una herramienta Web, llamada Scrum-UIA-MAT (**SCRUM-UIA** **MA**nagement **T**ool), que proporciona un sistema de gestión de proyectos basado en Scrum-UIA, y que considera diferentes funcionalidades para apoyar el desarrollo ágil y centrado en el usuario de la AI.

Respecto a la metodología propuesta, Scrum-UIA (punto 1), ésta prescribe en detalle 16 actividades, organizadas en los siguientes grupos de “Investigación Contextual”, “Gestión de Requisitos”, “Planificación del *Sprint*” e “Inspección y Mejora Continua”. Para cada actividad, se detallan las correspondientes tareas, los responsables y las técnicas o métodos para su realización. Del mismo modo, se detallan la información de entrada y los productos de salida requeridos por cada tarea. Se prescribieron un total de 29 tareas y 31 productos.

A continuación, se van a detallar los aportes concretos de la metodología Scrum-UIA:

- Para el grupo de actividades "Investigación Contextual", se define la “Actividad 1.1. Identificación y análisis del contexto”, donde se desarrolla el “Informe sobre las características principales del contexto” y el “Informe sobre las amenazas y oportunidades del contexto”. Dichos documentos describen las particularidades del entorno donde ocurre el proyecto, y las variables que son necesarias a considerar en el tratamiento del Sistema de Información (SI). También, se define la “Actividad 1.2. Identificación y análisis de contenidos”, donde se desarrolla el “Informe sobre la identificación de los contenidos del proyecto”, el “Modelo de contenido general del proyecto” y el “Informe sobre el análisis de las prioridades del contenido”, que determinan los contenidos que inicialmente están asociados al proyecto, con el fin de entender su alcance, origen y prioridades. Estos documentos forman la información de base para identificar las prioridades de la AI. Asimismo, se define la “Actividad 1.3. Identificación y análisis de usuarios finales”, donde se desarrolla el “Informe preliminar de los usuarios finales del proyecto y sus características” y el “Informe final sobre las necesidades de los usuarios finales”, que registran los rasgos y criterios de aceptación de los usuarios finales respecto al proyecto. Finalmente, se define la “Actividad 1.4. Generación del panorama general del proyecto”, donde se desarrolla el “Informe final sobre los elementos y los aspectos relevantes del proyecto” y el “Catálogo de criterios de aceptación de los usuarios finales”. Ambos resultados permiten configurar el panorama general del proyecto, y conducir el proceso de priorización de los requisitos de acuerdo a las prioridades de la usabilidad, de la

AI y del valor de negocio (esto es, una medida estándar de valor, utilizada para determinar la salud y el bienestar de la empresa en el largo plazo, es decir, el valor del mercado de una empresa). En resumen, se puede indicar que todas estas actividades proveen las bases para discutir, describir y considerar la perspectiva de los usuarios finales y las prioridades tanto de la AI como del proyecto, a través de todo el proceso de Scrum-UIA.

- Para el grupo de actividades "Gestión de Requisitos", se define la "Actividad 2.1. Identificación y análisis de requisitos del proyecto", donde se desarrolla el "Informe preliminar de requisitos del proyecto" y el "Catálogo de requisitos analizados del proyecto", donde se describen los requisitos funcionales, inicialmente identificados por el *Dueño del Producto* (Product Owner). Dichos informes proporcionan la información de base para configurar la generación final de los requisitos de la aplicación. De igual manera, en la "Actividad 2.2. Generación final de requisitos", se presenta la "*Lista del Producto* con los elementos descritos en formato de *Historia de Usuario*", el "Informe de la estimación de los elementos de la Lista del Producto" y el "Informe de las prioridades de los elementos de la *Lista del Producto*", que forman la *Lista del Producto* (Product Backlog) con los requisitos descritos (en formato de *historias de usuarios*), analizados (con la identificación de dependencias y conflictos), estimados (en *puntos de historias* y *días-persona*) y priorizados (a través de elementos cualitativos, relacionados con las prioridades del proyecto, es decir, mediante el enfoque del método de priorización propuesto). Un aspecto a destacar, es que el Arquitecto de la Información participa en esta actividad ("Actividad 2.2. Generación final de requisitos"), con el fin de ponderar con una mayor prioridad los elementos de la *Lista del Producto*, que evolucionan hacia el dominio de la solución y facilitan el desarrollo incremental en Scrum-UIA (en la técnica InterArch-T se ejemplifica como conducir el desarrollo incremental mediante los niveles de fidelidad<sup>1</sup> de los entregables de la AI). Continuando con la descripción del grupo de actividades "Gestión de Requisitos", se define la "Actividad 2.3. Determinación de la definición de *Terminado*", donde se desarrolla el "Informe sobre la definición de *Terminado*" para el *Sprint*". Este informe permite generar un entendimiento común sobre el alcance y las demandas de los requisitos presentados por el *Dueño del Producto* (definidos en la Actividad 2.2, en colaboración con el Arquitecto de la Información), y de las tareas que elaborará el *Equipo de Desarrollo* (Development Team) durante el *Sprint*. Finalmente, se define la "Actividad 2.4. Determinación del objetivo del *Sprint*", con un "Informe sobre el objetivo inicial para el *Sprint*", donde se determina un objetivo inicial para el siguiente *Sprint*, con el propósito de iniciar

---

<sup>1</sup>Existen diferentes formas para construir los entregables de la AI. El nivel de fidelidad puede variar para adaptarse a los propósitos del Arquitecto de Información, y de acuerdo a su complejidad se clasifican en diferentes niveles: desde un bosquejo en blanco y negro, para determinar en términos generales los espacios que ocupará cada elemento en el sitio Web (fidelidad baja), hasta representaciones avanzadas que incluyen mayor detalle visual (fidelidad alta), como imágenes, colores, tipografías, texto e iconos a utilizar.

la *Planificación del Sprint* (Sprint Planning) con un objetivo relacionado con las prioridades de los usuarios y de la AI. El objetivo final de estas actividades es identificar, analizar, priorizar y estimar los requisitos, en cada *Sprint* del proyecto. Estas actividades están dirigidas por el conocimiento obtenido en el grupo de actividades “Investigación Contextual”, con el fin de que los requisitos del proyecto reflejen el valor de negocio, las prioridades de los usuarios finales y del contenido.

- Para el grupo de actividades “Planificación del *Sprint*”, se define la “Actividad 3.1. Determinación de la capacidad del Equipo de Desarrollo”, donde se desarrolla el “Informe sobre la duración del *Sprint*”, el “Informe sobre la disponibilidad del Equipo de Desarrollo para el *Sprint*” y el “Informe sobre la velocidad del Equipo de Desarrollo”. Estos resultados permiten determinar el compromiso que el *Equipo de Desarrollo* puede asumir en el siguiente *Sprint*, y es la información de base para validar la consistencia de la estimación de las *tareas de desarrollo*. También, se define la “Actividad 3.2. Refinamiento del objetivo del *Sprint*”, donde se desarrolla el “Informe sobre el objetivo refinado del *Sprint*”. En dicho informe se determina el objetivo que finalmente se utilizará en el siguiente *Sprint* para guiar su ejecución. De igual modo, se define la “Actividad 3.3. Selección de requisitos para el *Sprint*”, donde se presenta la “*Lista de Pendientes del Sprint* con los requisitos para el *Sprint*”. Este resultado permite identificar todos los requisitos que se ejecutarán en el siguiente *Sprint*, que deben considerar las capacidades del *Equipo de Desarrollo*, determinadas en la actividad 3.1, y las prioridades definidas en la actividad 2.2. Asimismo, se define la “Actividad 3.4. Generación de tareas de desarrollo”, donde se desarrollan el “Informe de las tareas de desarrollo de los requisitos del *Sprint*”, el “Informe de las técnicas ágiles de la Arquitectura de la Información relacionadas con las tareas de desarrollo”, el “Informe sobre la estimación de las tareas de desarrollo” y el “Informe de los responsables de las tareas de desarrollo”. En conjunto, estos resultados forman la *Lista de Pendientes del Sprint* (Sprint Backlog) con sus elementos (*tareas de desarrollo*) estimados, asignados a responsables y asociados a técnicas ágiles de la AI. Un aspecto a destacar, es que el Arquitecto de la Información participa en esta actividad (“Actividad 3.4. Generación de tareas de desarrollo”), con el fin de identificar tareas de desarrollo relacionadas con la AI (*modelo de contenido*, *maquetas*, *planos*, entre otros) que permiten promover el desarrollo incremental en Scrum-UIA. Adicionalmente, en la “Actividad 3.4. Generación de tareas de desarrollo” se crea un “Diagrama de Clases UML” a partir de la elaboración del “Modelo de contenido general del proyecto” con la herramienta InterArch (este modelo de contenido es generado en la “Actividad 1.2. Identificación y análisis de contenidos”). De esta manera, el “Diagrama de Clases UML” se puede utilizar como fuente de información en actividades relacionadas con el diseño de la aplicación (*Ejecución del Sprint*). Finalmente, la “Actividad 3.5. Planificación de las tareas de desarrollo” genera un “Diagrama de la planificación diaria de las tareas de desarrollo”, donde se describe la estimación del esfuerzo diario de las

tareas de desarrollo durante la *Ejecución del Sprint*. Otro aspecto a destacar, es que el Arquitecto de la Información participa en esta actividad (“Actividad 3.5. Planificación de las tareas de desarrollo”), con el objetivo de asegurar que la organización de las tareas permita implementarlas de manera incremental, dirigida por los niveles de fidelidad de los entregables de la AI. En general, estas actividades tienen como objetivo identificar, estimar y planificar las tareas de desarrollo, que se requieren para implementar los requisitos (obtenidos desde el grupo de actividades “Gestión de Requisitos”), seleccionados para el *Sprint*.

- En cuanto al último grupo de actividades "Inspección y Mejora Continua", se define la “Actividad 4.1 Inspección de tareas de desarrollo”, donde se desarrolla el “Informe de la revisión de las tareas de desarrollo”. En este informe se examina el estado de las tareas (definidas en la actividad 3.4), que el *Equipo de Desarrollo* realiza (de acuerdo a la planificación determinada en la actividad 3.5) durante la *Ejecución del Sprint* y, a la vez, se inspecciona el cumplimiento de los criterios de aceptación de los usuarios finales. De igual manera, la “Actividad 4.2 Revisión del incremento del producto” presenta el “Informe sobre la revisión del incremento del producto”, donde se describe la evaluación del incremento del producto (es decir, parte del producto generado en el *Sprint* que está en condiciones de ser entregado), que se realiza mediante la participación directa de los usuarios finales. Finalmente, se define la “Actividad 4.3 Inspección de prácticas de trabajo”, donde se desarrolla el “Informe sobre la inspección de las prácticas de trabajo”, que identifica las observaciones respecto al desempeño de las prácticas y de las técnicas utilizadas para integrar la AI, con el fin de identificar las oportunidades de mejora relacionadas con los usuarios finales, y que se puedan abordar durante el siguiente *Sprint*. Este grupo de actividades tiene como objetivo principal examinar y evaluar las tareas de desarrollo y las prácticas de trabajo utilizadas en el *Sprint*, con el fin de fomentar el Diseño Centrado en el Usuario.

Por otro lado, tal y como se menciona en el grupo de actividades "Gestión de Requisitos", el proceso de priorización de los requisitos en Scrum-UIA se guía por elementos cualitativos relacionados con las prioridades del proyecto, identificados en el grupo de actividades “Investigación Contextual”. Para llevar a cabo lo anterior, se utiliza el enfoque del método de priorización propuesto, QMPSR (punto 2), que permite capturar formalmente las prioridades identificadas en Scrum-UIA y guiar el proceso de priorización de manera dinámica y ágil, para apoyar la integración de la AI centrada en los usuarios. QMPSR se compone de cuatro fases. En la primera fase (“Identificar los *aspectos* que dirigen la evaluación en el proceso de priorización”) se especifican y priorizan las características que los tomadores de decisiones consideran cuando priorizan los requisitos que tienen valor para el proyecto. En la segunda fase (“Identificar los *elementos* de los *aspectos*”) se identifican y priorizan los *elementos* para cada *aspecto* definido en la primera fase. En la tercera fase (“Priorización basada en los *elementos* de los *aspectos*”) se obtiene la priorización de los requisitos a través de la relación entre los requisitos y los *elementos* de los *aspectos* relevantes del proyecto

(definidos en la primera y segunda fases, respectivamente). Finalmente, en la última fase (“Computar el ranking final”) se genera la clasificación final de los requisitos, de acuerdo con el nivel de relevancia de cada requisito, considerando todos los *aspectos* del proyecto.

Como se ha señalado anteriormente, el Arquitecto de la Información participa en diferentes actividades (2.2, 3.4 y 3.5) para promover el desarrollo incremental de los requisitos en Scrum-UIA a través de la AI. Para alcanzar este objetivo, se utiliza la técnica InterArch-T (y su implementación mediante la herramienta CASE InterArch) que permite fomentar la continuidad en la implementación de los requisitos en Scrum-UIA, a partir de los productos relacionados con la AI. Además, la herramienta InterArch se puede utilizar tanto en las fases iniciales, como durante la *Ejecución del Sprint*, para integrar la AI y aumentar la automatización e interoperabilidad en el desarrollo de los productos. InterArch-T se compone de tres elementos. El primero es el “Modelado Visual de la AI”, que se lleva a cabo mediante la interfaz de InterArch y se compone de diferentes barras de herramientas para elaborar y gestionar los diagramas en un entorno funcional. El segundo componente corresponde a la “Transformación del Modelo Visual” que permite identificar los elementos visuales elaborados por el Arquitecto de la Información para transformarlos (utilizando cinco reglas de asociación y relación) en diagramas UML (Unified Modeling Language), utilizados por los miembros del *Equipo de Desarrollo* de Scrum-UIA (Analistas e Ingenieros del Software). Finalmente, el último componente corresponde a la “Generación Textual Transportable en Diagramas UML”, donde se toman como entrada las relaciones creadas en el componente anterior y se generan diagramas de clases UML en formato XML transportable, llamado XMI (XML Metadata Interchange), para los miembros del *Equipo de Desarrollo* de Scrum-UIA.

Tal y como se mencionó más arriba, Scrum-UIA incluye una serie de características específicas que intentan dirigir el desarrollo de las aplicaciones interactivas de manera ágil y centrada en los usuarios. Una implementación eficiente de las características de Scrum-UIA exige proveer la capacidad para organizar, planificar y estructurar los proyectos en un entorno digital integrado, que facilite y apoye la realización de las diferentes tareas demandadas. Para hacer frente a esta exigencia, se ha implementado la metodología Scrum-UIA y su método de priorización (QMPSR), mediante la construcción de la herramienta Web Scrum-UIA-MAT (punto 4). Scrum-UIA-MAT está compuesta por 16 módulos, agrupados en seis ramas de navegación: Proyecto, *Lista del Producto*, *Sprints*, Tareas, Planificación y Configuración. La rama de navegación “Proyecto” presenta los módulos “Consultar Proyectos (Mod-11)” y “Agregar Proyecto (Mod-12)” que permiten consultar y gestionar los proyectos software que se implementan, utilizando la metodología Scrum-UIA. En la rama de navegación “Lista del Producto”, se desarrollan los módulos “Consultar Historias de usuarios (Mod-21)”, “Gestionar Historias de usuarios (Mod-22)” y “Organizar Lista del Producto (Mod-23)”, permitiendo consultar, insertar, modificar, eliminar y priorizar los requisitos de cada proyecto, definido con el módulo “Agregar Proyecto (Mod-12)”. Cabe destacar que los módulos “Gestionar Historias de usuarios (Mod-22)” y “Organizar Lista del Producto



(Mod-23)” implementan el enfoque del método de priorización (QMPSR) propuesto para Scrum-UIA. En la rama de navegación “*Sprints*”, se desarrollan los módulos “Configurar *Sprints* (Mod-31)” y “Agregar *Sprint* (Mod-32)” que permiten establecer la configuración de cada *Sprint* y determinar los requisitos que los componen. La rama de navegación “Tareas” contiene los módulos “Consultar Tareas (Mod-41)” y “Gestionar Tareas (Mod-42)”, haciendo posible agregar, modificar, eliminar y consultar las tareas que realizará el *Equipo de Desarrollo* para implementar los requisitos seleccionados para el *Sprint*. De igual manera, la rama de navegación “Planificación” desarrolla el módulo de “Estimación del Esfuerzo de las Tareas (Mod-51)”, permitiendo estimar el esfuerzo diario de las tareas de desarrollo. Finalmente, la rama de navegación “Configuración” presenta los módulos “Opciones de Configuración (Mod-61)”, “Gestionar Usuarios Finales (Mod-62)”, “Gestionar Aspectos (Mod-63)”, “Gestionar Elementos (Mod-64)”, “Configurar Prioridades de los Aspectos (Mod-65)” y “Gestionar la Capacidad del Equipo (Mod-66)”. Estos módulos permiten agrupar las opciones de configuración de la herramienta, especificar los usuarios finales, determinar los *aspectos* relevantes del proyecto, identificar los *elementos* de los *aspectos*, definir las prioridades de los *aspectos* y determinar la capacidad del *Equipo de Desarrollo*, respectivamente. Entre éstos destacan los módulos “Gestionar Aspectos (Mod-63)” y “Gestionar Elementos (Mod-64)”, que permiten configurar los elementos cualitativos que se utilizan en el proceso de priorización de los requisitos, mediante el método de priorización QMPSR.

Asimismo, se realiza una verificación de la metodología Scrum-UIA a través de la aplicación de un ejemplo práctico, con el apoyo de Scrum-UIA-MAT, centrándose en la implementación de las actividades propuestas. Esta verificación consiste en la ejecución, de manera ordenada, de todas las tareas y actividades de Scrum-UIA, además de la creación de cada uno de los productos definidos, considerando como fuente del dominio del problema la construcción de un SI de gestión de aprendizaje para una universidad. La prescripción de las actividades propuestas en Scrum-UIA y su verificación mediante un escenario de prueba con el soporte de Scrum-UIA-MAT permiten ilustrar la idoneidad de las actividades prescritas para integrar la AI dentro de un modelo de proceso ágil de desarrollo centrado en el usuario.

Finalmente, se realizan distintos experimentos para evaluar QMPSR, InterArch y Scrum-UIA-MAT.

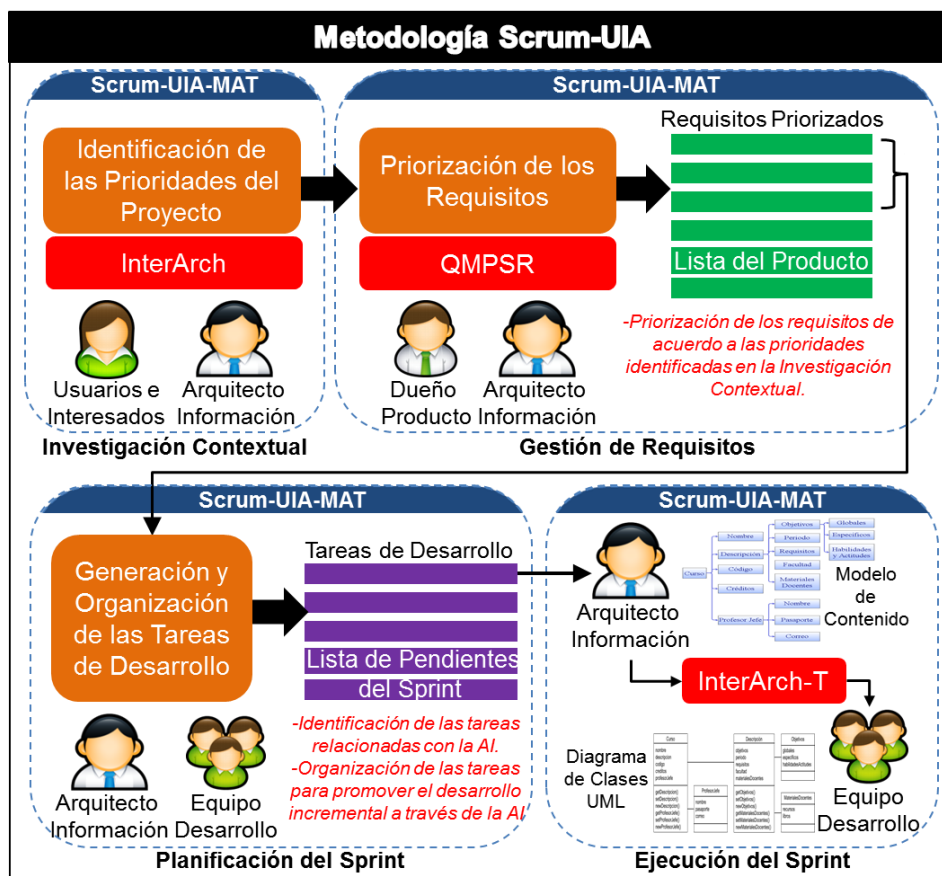
La primera evaluación consiste en estudiar las *colisiones de requisitos*, generadas por el método de priorización QMPSR. Para esta evaluación, se ha creado un marco experimental, que permite realizar evaluaciones con diferentes métodos de priorización, en base a características específicas. En este marco experimental, se definen las métricas de *esfuerzo de priorización* y de *colisión de requisitos*, para realizar nueve experimentos con diferentes conjuntos de requisitos de entrada, comparando QMPSR con seis conocidos métodos de priorización existentes (MoSCoW, Orientado al Valor, Weiger, Definición de Producto, AHP y Kano). Los resultados demuestran que QMPSR supera uniformemente a todos los métodos de priorización comparados, generando menos

*requisitos colisionados* y presentando un buen comportamiento en términos de *escalabilidad*.

La segunda evaluación consiste en medir el grado de usabilidad de la herramienta InterArch, mediante un diseño cuasi experimental con 12 Arquitectos de la Información. Los resultados obtenidos muestran opiniones positivas y aceptables sobre la percepción del usuario en cuanto a la *Utilidad*, *Facilidad de Uso*, *Facilidad de Aprendizaje* y *Satisfacción*. Esto permite concluir que la herramienta InterArch es fácil de usar y de aprender, proporcionando, además, continuidad entre la información de análisis conceptual de la AI y la información de análisis y diseño que necesitan los Analistas e Ingenieros del Software.

Por último, la tercera evaluación consiste en medir el grado de usabilidad de la herramienta Scrum-UIA-MAT mediante un diseño cuasi experimental con 16 Ingenieros del Software. Una vez más, los resultados obtenidos muestran opiniones positivas y aceptables sobre la percepción del usuario en cuanto a la *Utilidad*, *Facilidad de Uso*, *Facilidad de Aprendizaje* y *Satisfacción*. Esto permite concluir que la herramienta Scrum-UIA-MAT es fácil de usar y de aprender por parte de sus usuarios potenciales, y que provee diversas funcionalidades para facilitar el acceso y la recolección de información de las distintas etapas de la gestión de proyectos software.

Como resumen de la solución propuesta, se ha elaborado la Figura 1 para presentar e identificar gráficamente el uso de sus diferentes componentes.



**Figura 1:** Solución propuesta.

Como se puede observar en la Figura 1, en la fase inicial de Scrum-UIA (*Investigación Contextual*) se identifican las diferentes prioridades relacionadas con el proyecto. En esta fase, el Arquitecto de la Información elabora diferentes artefactos, con el fin de proveer un panorama general del proyecto. En este contexto, se genera un *modelo de contenido* general del proyecto, mediante el uso de la herramienta InterArch, para facilitar la discusión sobre la organización y la gestión de los contenidos. Esto permite crear, de manera automática, información de diseño (*diagrama de clases UML*) que se puede utilizar en fases posteriores del proyecto (*Planificación y Ejecución del Sprint*) por los miembros del *Equipo de Desarrollo*.

Por otro lado, el proceso de priorización de los requisitos en Scrum-UIA (ver *Gestión de Requisitos* en la Figura 1) se lleva a cabo mediante el enfoque propuesto en el método de priorización QMP SR. Este método considera formalmente las prioridades relacionadas con la AI, la usabilidad y el valor de negocio a través del uso de *elementos y aspectos* cualitativos.

De este modo, los requisitos con mayor prioridad se seleccionan para el *Sprint*, donde el Equipo de Desarrollo define las tareas específicas de desarrollo para llevar a cabo (ver *Planificación del Sprint* en la Figura 1). El Arquitecto de la Información participa en esta actividad, con el fin de identificar las tareas relacionadas con la AI y organizarlas para promover el desarrollo incremental, de acuerdo a los entregables de la AI.

Tras la *Planificación del Sprint*, la *Ejecución del Sprint* se conduce de acuerdo al enfoque de la técnica InterArch-T, y las tareas de desarrollo relacionadas con la AI se elaboran con la herramienta InterArch, con el fin de proveer continuidad y reducir el esfuerzo en la implementación de los requisitos en Scrum-UIA.

Adicionalmente, la metodología Scrum-UIA cuenta con el soporte tecnológico de la herramienta Scrum-UIA-MAT. Esta herramienta permite recolectar, organizar y gestionar la información de las distintas actividades y tareas de Scrum-UIA.

### 1.6. Fortalezas

Conforme a la solución propuesta y a los objetivos planteados, a continuación se presentan algunas de las fortalezas más importantes que esta investigación promueve:

- *Permite la incorporación y participación de los usuarios finales en un proceso ágil de desarrollo de software.*

Las metodologías ágiles tradicionales están orientadas principalmente a los clientes y el valor de negocio, más que a los usuarios finales. No obstante, la metodología propuesta permite incorporar y tener en cuenta las necesidades y las prioridades de los usuarios finales durante todo el ciclo de vida de desarrollo de software, a través de la identificación de diferentes instancias de participación e involucración de ellos. Además, los diversos entregables de valor (incrementos de producto), generados en cada *Sprint*, son validados con la participación directa de los usuarios finales, lo que permite fortalecer su retroalimentación y participación continua.

- *Permite utilizar elementos cualitativos para argumentar y validar la prioridad de los requisitos.*

Los métodos de priorización de requisitos actuales no proveen una comprensión global de las preferencias del proyecto y de los usuarios finales, que permita guiar, de manera dinámica, el proceso de priorización y verificar la consistencia de las evaluaciones de los requisitos. Sin embargo, esta tesis introduce un nuevo método de priorización, que permite capturar, de manera colectiva y dinámica, las principales prioridades que se deben considerar para discernir el nivel de relevancia de los requisitos. Estas prioridades son capturadas formalmente, y permiten verificar y discutir la exactitud y coherencia de las evaluaciones de la clasificación final de los requisitos.

- *Permite promover el desarrollo incremental dirigido por la Arquitectura de la Información.*

Esta tesis introduce una técnica para representar los modelos de contenido, elaborados por el Arquitecto de la Información en fases iniciales del proyecto, que se transforman, mediante un conjunto de reglas de relación y asociación, en descripciones más cercanas al dominio de la solución, es decir, en diagramas de clases UML para Analistas e Ingenieros del Software. Esta información permite dar continuidad a la información conceptual de la AI y la información de análisis y diseño que requieren los Analistas e Ingenieros del Software, promoviendo así la implementación de los requisitos de manera incremental.

- *Permite gestionar los proyectos software en un entorno integrado y fácil de usar.*

Esta tesis propone una herramienta Web que permite sistematizar la metodología propuesta, así como su método de priorización (QMPSR), y provee diferentes funcionalidades para gestionar proyectos software y apoyar la integración de la AI en un marco ágil de desarrollo centrado en el usuario. Algunas de estas funcionalidades corresponden a: provee opciones para organizar la *Lista del Producto*, de acuerdo a las preferencias del *Dueño del Producto*, el Arquitecto de la Información y los *aspectos* relevantes del proyecto; permite gestionar los requisitos de cada proyecto, considerando la perspectiva de los usuarios finales (identificación de los usuarios finales, del valor de negocio, de los criterios de aceptación); proporciona sugerencias sobre actividades y técnicas para integrar la AI en entornos ágiles, entre otras.

## 1.7. Debilidades

En esta sección se detallan algunos de los aspectos no cubiertos y debilidades de la tesis doctoral que, si bien no se han implementado por razones de extensión y tiempo de realización de esta investigación, serán considerados como posibles ampliaciones y trabajo futuro:

- *No se provee una validación de las actividades y las tareas de Scrum-UIA a través de un entorno real de una empresa de desarrollo de software.*

Para la verificación de las actividades prescritas en la metodología Scrum-UIA, sólo se realiza un desarrollo de un ejemplo práctico mediante una aplicación determinada. Esta verificación se centra en la ejecución de las actividades de Scrum-UIA junto con el soporte de la herramienta Scrum-UIA-MAT. Sin embargo, sería conveniente validar las actividades propuestas mediante la construcción de diferentes aplicaciones interactivas en un proyecto real, dentro de un entorno de una empresa de desarrollo. Cabe destacar que la aplicación de un ejemplo práctico de desarrollo permite ilustrar y aclarar la idoneidad de las tareas, las actividades y los productos propuestos para facilitar la integración de la AI dentro un proceso ágil de desarrollo centrado en el usuario. Asimismo, este ejemplo concreto puede servir de base para guiar los aspectos generales de una futura implementación del modelo de proceso de Scrum-UIA en un proyecto real.

- *Las actividades de inspección y mejora continua de Scrum-UIA no cuentan con el soporte de la herramienta Scrum-UIA-MAT.*

La herramienta Scrum-UIA-MAT no tiene funcionalidades para registrar los resultados de las actividades de inspección y mejora continua de Scrum-UIA (*Scrum Diario*, *Revisión del Sprint* y *Retrospectiva del Sprint*). Esto se debe a que, principalmente, el desarrollo de las funcionalidades en Scrum-UIA-MAT se ha centrado en sistematizar las actividades de Scrum-UIA que permiten dar continuidad al proceso de desarrollo. No obstante, la elaboración de módulos específicos para sistematizar las actividades de inspección y mejora continua de Scrum-UIA es un aspecto importante a considerar en futuras mejoras de la herramienta.

- *La técnica propuesta para promover el desarrollo incremental no provee soporte para todos los entregables de la Arquitectura de la Información.*

La herramienta CASE InterArch, que implementa la técnica InterArch-T para promover el desarrollo incremental en Scrum-UIA, se concentra en transformar los *modelos de contenido* (elaborados por los Arquitectos de la Información) en diagramas de clases UML (requeridos por Analistas e Ingenieros del Software). Esto sólo corresponde a uno de los entregables de la AI, sin considerar a otros, tales como *los planos*, *las maquetas* y *los vocabularios controlados*. Esto es debido a que, principalmente, no se ha encontrado una correspondencia directa que permita generar información de forma automática con otros productos, requeridos por Analistas o Ingenieros del software. No obstante, el tratamiento de los *modelos de contenido* corresponde a un avance importante en cuanto a la automatización e interoperabilidad entre las definiciones conceptuales de los contenidos de una aplicación Web, y los elementos de análisis y diseño procesables por Analistas e Ingenieros del Software.

## 1.8. Estructura de la Tesis Doctoral

El trabajo llevado a cabo se ha estructurado de la siguiente manera:

- En el **Capítulo 1** se han presentado los fundamentos principales de la investigación. De este modo, se ha identificado el problema que afronta la investigación, las hipótesis de partida y las preguntas de investigación en las que se apoya la tesis. También, se ha descrito la metodología de trabajo para corroborar las hipótesis de partida. Además, se han introducido los objetivos y la solución propuesta, así como sus fortalezas y debilidades.
- En el **Capítulo 2** se presenta el estado del arte. En la sección inicial se realiza un estudio sistemático de la literatura, con el fin de buscar contribuciones concretas que aborden la problemática identificada. A continuación, se analizan las diferentes propuestas para la AI, con el fin de identificar si presentan un modelo de proceso ágil de desarrollo centrado en el usuario. De igual manera, a fin de desplegar una sólida AI, se estudian las metodologías ágiles para identificar en qué medida permiten integrar el Diseño Centrado en el Usuario. Similarmente, se identifican las prácticas y las recomendaciones utilizadas para integrar eficientemente el Diseño Centrado en el Usuario dentro del proceso ágil de desarrollo. En último lugar, se analiza la capacidad de los métodos de priorización actuales para conducir el proceso de priorización a través de las prioridades de la usabilidad, de la AI y del valor de negocio.
- En el **Capítulo 3** se presenta en detalle la solución propuesta, describiendo sus diferentes componentes. Se definen y prescriben las actividades de la metodología Scrum-UIA. Asimismo, se presenta el método de priorización de requisitos QMPSR, creado para la metodología Scrum-UIA. Posteriormente, se describe la técnica InterArch-T, creada para promover el desarrollo incremental de los requisitos en Scrum-UIA a través de la AI, así como su implementación mediante la herramienta CASE InterArch. Finalmente, se introduce la herramienta Web Scrum-UIA-MAT, ideada para sistematizar la aplicación de la metodología Scrum-UIA.
- En el **Capítulo 4** se describe la verificación realizada para la metodología Scrum-UIA. Esta verificación consiste en la aplicación de un ejemplo práctico de desarrollo, utilizando el soporte de la herramienta Scrum-UIA-MAT y considerando como fuente del dominio del problema la construcción de un SI de gestión de aprendizaje para una universidad.
- En el **Capítulo 5** se detallan las evaluaciones realizadas sobre QMPSR, InterArch y Scrum-UIA-MAT. Inicialmente, se describen los experimentos ejecutados para evaluar y comparar las *colisiones de requisitos*, generadas en QMPSR. Más adelante, se describe la evaluación de la usabilidad de InterArch con 12 Arquitectos de la Información. Finalmente, se presenta la experimentación realizada para medir el grado de usabilidad de la herramienta Scrum-UIA-MAT con 16 Ingenieros del Software. Para cada una de las evaluaciones, se provee una descripción detallada de todo el proceso

experimental realizado, incluyendo un análisis y una discusión sobre los resultados obtenidos.

- En el **Capítulo 6** se presentan las conclusiones relacionadas con la investigación presentada en esta tesis doctoral. De igual manera, se exponen aspectos a abordar como trabajo futuro, con el fin de dar continuidad a la investigación realizada.

## 1.9. Publicaciones

A continuación, se indican las publicaciones a las que ha dado lugar esta tesis doctoral, agrupadas de acuerdo a su categoría:

### ➤ JOURNALS CON ÍNDICE DE IMPACTO JCR

- **Rojas, L. A.** and Macías, J. A. “Bridging the Gap between Information Architecture Analysis and Software Engineering in Interactive Web Application Development”. Science of Computer Programming. Elsevier. Vol. 78, Issue 11, November 2013, pp. 2282-2291. ISSN: 0167-6423. DOI: 10.1016/j.scico.2012.07.020.

#### **Criterios de Calidad:**

- Factor de impacto JCR Science Edition año 2013: 0,548. Puesto 86/105, Q4 (Computer Science, Software Engineering).
- SCImago Journal Rank (SJR) año 2013: 0,721, Q2 (Software).
- Google Scholar Metrics (journal): h5-index: 27, h5-median: 37.
- **Rojas, L. A.** and Macías, J. A. “Sistema Automatizado de Integración de Arquitectura de la Información en el Desarrollo de Aplicaciones Web Interactivas”. El Profesional de la Información. Volumen 21, no. 2, 2012, pp. 160-165 ISSN: 1386-6710. DOI: 10.3145/epi.2012.mar.06.

#### **Criterios de Calidad:**

- Factor de impacto JCR Social Science Edition año 2012: 0,439. Puesto 53/85, Q3 (Information Science y Library Science).
- SCImago Journal Rank (SJR) año 2012: 0,310, Q2 (Communication).
- Índice de Impacto de Revistas Españolas de Ciencias Sociales (IN-RECS). Posición 1/33, Q1 (Documentación).
- Revista internacional científica y profesional en español líder en el área de información, comunicación, documentación, bibliotecas y nuevas tecnologías.
- Primera revista española del sector indexada por las dos bases de datos bibliográficas internacionales más importantes: ISI Social Science Citation Index y Scopus. Además de Inspec, Lisa, Lista, Pascal, Francis, Isoc, etc.

- Google Scholar Metrics (journal): h5-index: 20, h5-median: 30.

#### ➤ JOURNALS SIN ÍNDICE DE IMPACTO JCR

- **Rojas, L. A.** y Macías, J. A. “Soporte para el Análisis de la Arquitectura de la Información en Aplicaciones Web”. *Faz*, no. 5, enero de 2012, pp. 10-27. ISSN 0718-526X.

##### **Criterios de Calidad:**

- Revista que cuenta con comité científico internacional y contiene sólo artículos de investigación científica.
- Artículo sometido a evaluación externa por pares.
- Revista con el 100% de los artículos con autores no vinculados con la institución editora.

#### ➤ CONGRESOS INTERNACIONALES

- **Rojas, L. A.** and Macías, J. A. “An Agile Information-Architecture-Driven Approach for the Development of User-Centered Interactive Software”. *Proceedings of the 16th International Conference on Human Computer Interaction (Interacción 2015)*. Pere Ponsa and Daniel Guasch (eds.) Vilanova i la Geltrú, Barcelona, Spain. 7-9 September 2015. Article No. 50. ICPS: ACM International Conference Proceedings Series. ACM New York, USA, 2015. ISBN: 978-1-4503-3463-1. DOI: 10.1145/2829875.2829919.

##### **Criterios de Calidad:**

- Indexado en Scopus, ISI Web of Knowledge, DBLP y ACM Digital Library.
  - Artículo sometido a evaluación externa por pares.
  - Google Scholar Metrics (congreso): h5-index: 5, h5-median: 7.
- **Rojas, L. A.** and Macías, J. A. “End-User Support for Information Architecture Analysis in Interactive Web Applications”. *13th IFIP TC 13 International Conference (Interact 2011)*. Campos, Nunes, Graham, Jorge and Palanque (editors). Lisbon, Portugal, September 5-9, 2011, pp. 515-518. *Lecture Notes in Computer Science*, Volume 6949, Springer NY, 2011, pp. 515-518. ISSN: 0302-9743. ISBN: 978-3-642-23767-6. DOI: 10.1007/978-3-642-23768-3\_71.

##### **Criterios de Calidad:**

- Congreso catalogado como CORE A (Field of Research: Information Systems).
- Indexado en Scopus, Springer Link, ACM Digital Library y IFIP.
- Google Scholar Metrics (congreso): h5-index: 25, h5-median: 35.



➤ **CONGRESOS INTERNACIONALES DE ÁMBITO IBEROAMERICANO**

- **Rojas, L. A.** y Macías, J. A. “Soporte para el Análisis de la Arquitectura de la Información en Aplicaciones Web”. Actas del XII Congreso Internacional de Interacción Persona Ordenador (Interacción 2011). Nestor Garay Vitoria y Julio Abascal González (Editores). Ibergarceta Publicaciones. Lisboa, Portugal, del 02 al 05 de Septiembre de 2011, pp. 89-98. ISBN: 978-84-9281-234-9.

**Criterios de Calidad:**

- Artículo sometido a evaluación externa por pares.



# Capítulo 2: Estado del Arte

---

**E**n este capítulo, se presentan los trabajos relacionados con la investigación. En la sección 2.1, se realiza un estudio sistemático de la literatura, con el fin de corroborar la hipótesis de partida **H1.1** y justificar la realización de la presente tesis doctoral. En la sección 2.2, se examinan las diferentes propuestas para la Arquitectura de la Información, con el objetivo de analizar en profundidad sus principales características y corroborar la hipótesis de partida **H1.2**. En la sección 2.3, se analizan las metodologías ágiles, con el fin de identificar en qué medida éstas permiten integrar el Diseño Centrado en el Usuario y corroborar la hipótesis de partida **H1.3**. En la sección 2.4, se analizan las diferentes prácticas y recomendaciones que se utilizan comúnmente para integrar el Diseño Centrado en el Usuario en las metodologías ágiles, con el objetivo de corroborar la hipótesis de partida **H1.4**. En la sección 2.5, se describen las limitaciones de los métodos de priorización para considerar, de manera dinámica, las preferencias de los proyectos y corroborar la hipótesis de partida **H1.5**. Finalmente, en la sección 2.5, se especifican las conclusiones más importantes de este capítulo.

## 2.1. Mapping Study

El tema de investigación de esta tesis doctoral se forma a partir del planteamiento de la hipótesis de partida **H1.1**: No existen trabajos anteriores que realicen una integración de la Arquitectura de la Información dentro de un proceso ágil de desarrollo centrado en el usuario. Para la corroboración de esta hipótesis de partida, se ha llevado a cabo un estudio sistemático de la literatura a través de la técnica Systematic Mapping Study (SMS) (Petersen et al., 2008), con el objetivo de identificar el estado actual de la literatura, respecto a las metodologías que permiten integrar la Arquitectura de la Información a través de un modelo de proceso ágil de desarrollo centrado en el usuario.

Asimismo, con la realización de este estudio también se busca responder a la siguiente pregunta de investigación (planteada anteriormente en la sección 1.2):

**PI<sub>1</sub>**: ¿Qué propuestas existen para integrar la Arquitectura de la Información dentro de un modelo de proceso ágil de desarrollo centrado en el usuario?

### 2.1.1. Parámetros

Para llevar a cabo el estudio sistemático de la literatura, inicialmente se han definido y determinado un conjunto de parámetros, para establecer los lineamientos principales de este estudio. A continuación, se describe cada uno de estos parámetros en los siguientes apartados.

#### 2.1.1.1. Objetivo

El principal objetivo del estudio es determinar cuántas publicaciones se relacionan con el tema investigado, y cuál es la relevancia de cada una de ellas.

### 2.1.1.2. Literatura Revisada

La literatura revisada consiste en publicaciones indexadas en las siguientes bases de datos:

- IEEE Xplore
- ACM Digital Library
- Springerlink
- ScienceDirect
- ISI Web of Knowledge
- Scopus

### 2.1.1.3. Áreas de Análisis

Para lograr el objetivo de este estudio, se definieron diferentes áreas de análisis, con el fin de tener un marco común para comparar y analizar las publicaciones. El criterio para realizar el análisis se centra principalmente en cuatro áreas:

- **Área 1 - Modelos de Proceso:** Identificar los modelos de proceso o metodologías de desarrollo utilizadas para llevar a cabo la AI.
- **Área 2 - Enfoque Centrado en el Usuario:** Identificar los enfoques, actividades, herramientas y técnicas, orientadas al Diseño Centrado en el Usuario y utilizadas para llevar a cabo la AI.
- **Área 3 - Enfoque Ágil:** Identificar los enfoques, actividades, herramientas y técnicas, orientadas al diseño ágil que se utilizan para desplegar la AI.
- **Área 4 - Despliegue de la AI:** Analizar el grado de integración de los modelos de proceso, los enfoques centrados en el usuario y ágiles, y si las propuestas están explícitamente personalizadas para la AI.

### 2.1.1.4. Cadena de Búsqueda

Se definió una cadena de búsqueda, con el objetivo de abarcar las cuatro áreas de análisis, descritas en la sección 2.1.1.3:

```
("information architecture" OR "architecture of information")  
AND  
("usability" OR "life cycle" OR "human centred" OR "user  
centered" OR "user centric" OR "user centred" OR "methodology"  
OR "process" OR "model" OR "development") AND  
("agile" OR "rapid" OR "XP" OR "extreme programming" OR "Scrum"  
OR "Adaptive Software Development" OR "Feature-Driven  
Development" OR "FDD" OR "Kanban" OR "Dynamic Systems  
Development Method" OR "DSDM")2
```

---

<sup>2</sup>Se han listado las metodologías ágiles que se indican de forma más recurrente en los documentos más representativos de la literatura ágil.

### 2.1.1.5. Criterios de Evaluación

Se evaluó cada uno de los documentos que resultaron de la búsqueda en las diferentes bases de datos, a partir de la cadena de búsqueda ideada. De esta forma, se eliminaron todas las repeticiones, pues muchos artículos estaban indexados en más de uno de esos índices. Posteriormente, se les asignó como criterio de evaluación un valor entre 0 y 3 en cada una de las cuatro áreas de análisis, definidas en la sección 2.1.1.3. Los valores asignados para los criterios de evaluación corresponden a: 0 para los documentos *Sin Relevancia*, 1 para los documentos con *Relevancia Baja*, 2 para los documentos con *Relevancia Media*, y 3 para los documentos con *Relevancia Alta*. En la Tabla 2, se describen en detalle los criterios de evaluación (0, 1, 2 y 3) que deben cumplir los documentos para cada una de las cuatro áreas de análisis (Modelos de Proceso, Enfoque Centrado en el Usuario, Enfoque Ágil y Despliegue de la AI).

Rel.	Área 1: Modelos de Proceso	Área 2: Enfoque Centrado en el Usuario	Área 3: Enfoque Ágil	Área 4: Despliegue de la AI <sup>3</sup>
Sin Relevancia (0)	No se indica el uso o propuesta de ningún modelo de proceso o metodología de desarrollo, ni de ninguna de sus fases.	No se indica ningún enfoque centrado en el usuario.	No se indica ningún enfoque ágil.	No se relaciona con la AI.
Relevancia Baja (1)	Se menciona el uso o propuesta de algún enfoque o marco de trabajo para el desarrollo de un proyecto Web. No obstante, los modelos de proceso no son el tema principal del estudio o no se describen actividades ni fases.	Se menciona algún enfoque centrado en el usuario. No obstante, no es el tema central del estudio, y no se indica que el enfoque se centre en el diseño o en el despliegue de la AI.	Se indica algún enfoque ágil. Sin embargo, no es el tema central del estudio, y no especifica que el enfoque se centre en el diseño o en el despliegue de la AI.	Se relaciona con la AI. No obstante, sólo una de las áreas de análisis se centra en la AI o se menciona de forma vaga.

<sup>3</sup>Aunque todos los documentos se evalúan de acuerdo a las áreas de análisis (Modelos de Proceso, Enfoque Centrado en el Usuario y Enfoque Ágil), cada documento también se evalúa de forma general, con el objetivo de determinar si el estudio permite integrar la AI considerando un enfoque ágil y centrado en el usuario.

Relevancia Media (2)	Se indica el uso o propuesta de algún enfoque o marco de trabajo para el desarrollo Web, siendo éste el tema principal del estudio, especificando actividades o fases en concreto. Sin embargo, aunque se analiza el desarrollo de un proyecto, no se analiza explícitamente el despliegue de la AI.	Se indica el uso de un enfoque centrado en el usuario, siendo éste el tema principal del estudio. Sin embargo, no se especifica que el enfoque se centra en el diseño o en el despliegue de la AI.	Se indica el uso de algún enfoque ágil, siendo éste el tema central del estudio. No obstante, no se especifica que el enfoque se centra en el diseño o en el despliegue de la AI.	El documento se relaciona con la AI. Sin embargo, esto sucede sólo en dos áreas de análisis.
Relevancia Alta (3)	Se indica el uso o propuesta de algún modelo de proceso o metodología de despliegue de la AI, o alguna de sus fases en concreto, siendo éste el tema principal del trabajo de investigación.	Se indica el uso de un enfoque centrado en el usuario y se especifica el enfoque en el diseño o en el despliegue de la AI, o en alguna de sus actividades, siendo éste el tema central del trabajo de investigación.	Se indica el uso de un enfoque ágil y se especifica el enfoque en el diseño o en el despliegue de la AI, o en alguna de sus actividades, siendo éste el tema central del trabajo de investigación.	El documento se relaciona con la AI. Además, todas las áreas de análisis se centran en ella.

**Tabla 2:** Criterios de evaluación para examinar cada documento en las áreas de análisis.

Para ejemplificar la aplicación de los criterios de evaluación, definidos en la Tabla 2, se ha elaborado la Tabla 3, para mostrar una aplicación real de los criterios de evaluación sobre el trabajo *“Information Architecture: For the Web and Beyond”* (Rosenfeld et al., 2015). Como se puede observar, la Tabla 3 presenta la justificación entregada para asignar el criterio de evaluación en cada una de las cuatro áreas de análisis.

Áreas de Análisis	Evaluación	Justificación
Área 1: Modelos de proceso	<i>Relevancia Alta</i> (3)	El trabajo presenta una metodología completa para la AI.
Área 2: Enfoque centrado en el usuario	<i>Relevancia Media</i> (2)	El trabajo presenta algunas técnicas y herramientas para integrar la AI centrada en el usuario, sin embargo, el tema principal no es el Diseño Centrado en el Usuario.
Área 3: Enfoque ágil	<i>Sin Relevancia</i> (0)	El trabajo no presenta un enfoque ágil.
Área 4: Despliegue de la AI	<i>Relevancia Media</i> (2)	El trabajo se relaciona directamente con la AI, proponiendo una metodología, y la utilización de un enfoque centrado en el usuario.

**Tabla 3:** Ejemplo de aplicación de los criterios de evaluación.

#### 2.1.1.6. Definición de Categorías para Filtrar Documentos Importantes

En esta sección se definen diferentes tipos de categorías (*Importante*, *Amplio*, *General* y *No Seleccionado*) para filtrar los documentos más importantes que se evaluaron anteriormente con los criterios de evaluación (sección 2.1.1.5). El objetivo de lo anterior es poder contar con un conjunto de documentos, cuyo contenido se analizará posteriormente en profundidad. La Tabla 4 presenta en detalle las cuatro categorías y los criterios que deben cumplir los documentos. Cabe destacar que los criterios son excluyentes, con el fin de clasificar los documentos en una sola categoría.

Categoría	Descripción	Criterio
<i>Importante</i>	Se seleccionan todos los documentos que son considerados con una <i>Relevancia Alta</i> en alguna de las áreas de análisis de la investigación.	Obtener una valoración de <i>Relevancia Alta</i> (3) en alguno de los criterios de evaluación.
<i>Amplio</i>	Se seleccionan todos los documentos, cuyo estudio considere todas las áreas de análisis de la investigación con algún grado de relevancia.	Obtener algún grado de relevancia (1 o 2) en todas áreas de análisis.
<i>General</i>	Se seleccionan todos los documentos, cuyo estudio tenga una <i>Relevancia Media</i> (2) en al menos tres de los criterios de evaluación.	Obtener una suma de la valoración de todos criterios de evaluación igual o mayor a 6.

<i>No Seleccionado</i>	No se seleccionan los documentos que no cumplan con alguno de los criterios, definidos anteriormente (importante, amplio y general).	No obtiene ninguna de las categorías anteriores (importante, amplio y general).
------------------------	--	---

**Tabla 4:** Categorías para filtrar documentos importantes.

Para ejemplificar la aplicación de los criterios de las categorías para filtrar documentos importantes, definidos en la Tabla 4, se ha elaborado la Tabla 5 que muestra una aplicación real en cuatro trabajos diferentes.

Documentos	Área 1: Modelos de Proceso	Área 2: Enfoque Centrado en el Usuario	Área 3: Enfoque Ágil	Área 4: Despliegue de la AI	Resultado
“LUCIA: Development of a comprehensive Information Architecture process model for websites” (Reichenauer, 2005)	3	3	0	2	<i>Importante</i>
“Development of information architecture for user-centered hospital KIOSK” (Hee et al., 2011)	1	1	1	2	<i>Amplio</i>
“Pro iOS Web design and development: HTML5, CSS3, and JavaScript with Safari” (Picchi, 2011)	2	0	2	2	<i>General</i>



“Enhanced approach for developing Web applications using model driven architecture” (Mukhtar et al., 2013)	0	0	0	0	No Seleccionado
--	---	---	---	---	-----------------

**Tabla 5:** Ejemplos de aplicación de los criterios de evaluación.

Como se puede observar en la Tabla 5, el documento “*LUCIA: Development of a comprehensive information architecture process model for websites*” obtiene una evaluación de *Relevancia Alta* (3) en las áreas sobre *Modelos de Proceso* y *Enfoque Centrado en el Usuario* (ver sección 2.1.1.3). Por lo tanto, este documento obtiene la categoría de *Importante*, ya que cumple con el criterio de “*obtener una valoración de Relevancia Alta (3) en alguno de los criterios de evaluación*” (ver Tabla 4).

Por otro lado, el documento “*Development of information architecture for user-centered hospital KIOSK*” (Hee et al., 2011) obtiene algún grado de relevancia (*Relevancia Baja* o *Relevancia Media*) en todas las áreas de análisis. Este documento obtiene la categoría de *Amplio*, ya que cumple con el criterio de “*obtener algún grado de relevancia (1 o 2) en todas áreas de análisis.*”.

Asimismo, el documento “*Pro iOS Web design and development: HTML5, CSS3, and JavaScript with Safari*” (Picchi, 2011) obtiene una *Relevancia Media* (2) en tres áreas de análisis (*Modelos de Proceso*, *Enfoque Centrado en el Usuario*, *Enfoque Ágil y Despliegue de la AI*). Por lo tanto, este documento obtiene la categoría de *General*, ya que cumple con el criterio de “*obtener una suma de la valoración de todos criterios de evaluación igual o mayor a 6*”.

Finalmente, el documento “*Enhanced approach for developing Web applications using model driven architecture*” (Mukhtar et al., 2013) es evaluado como *Sin Relevancia* en las cuatro áreas de análisis. De esta manera, este documento no obtiene ninguna de las categorías definidas para filtrar documentos importantes y, por lo tanto, no es seleccionado.

En resumen, todos los documentos que obtienen alguna categoría (*Importante*, *Amplio* o *General*), se seleccionan para posteriormente analizarlos en profundidad. Por el contrario, si el documento no obtiene ninguna categoría es descartado, como es el caso de “*Enhanced approach for developing Web applications using model driven architecture*” (Mukhtar et al., 2013).

#### 2.1.1.7. Proceso del Estudio Sistemático de la Literatura

Teniendo en cuenta los parámetros anteriormente descritos, el proceso general para la selección de los documentos relevantes del estudio se divide en seis pasos, tal y como se

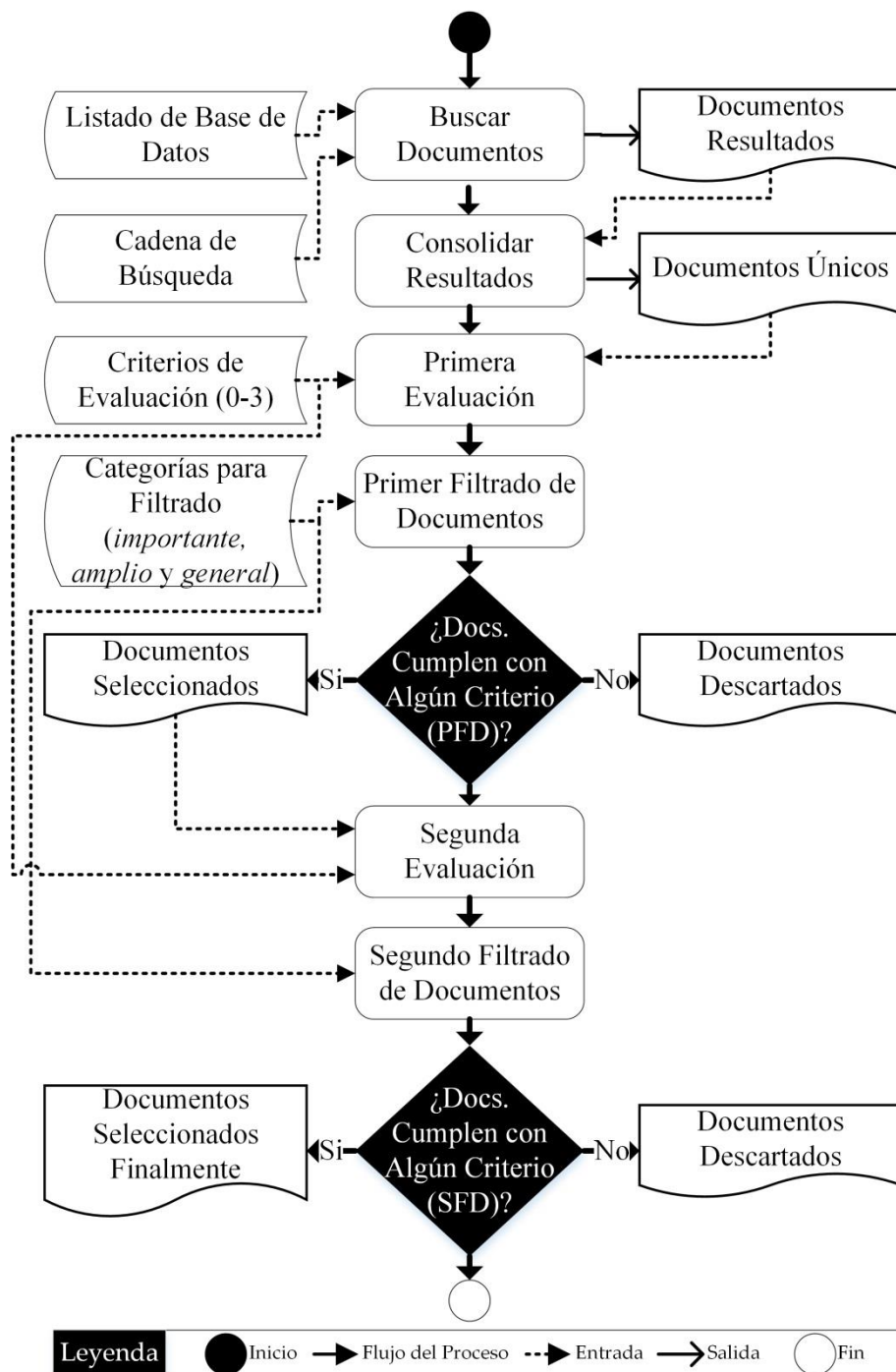
muestra en la Figura 2. A continuación, se describe en detalle cada uno de los pasos del proceso:

- **Buscar Documentos:** En este paso se realiza la búsqueda de los documentos en las diferentes bases de datos que fueron definidas en la sección 2.1.1.2. Asimismo, la búsqueda se realiza de acuerdo a la cadena de búsqueda, definida en la sección 2.1.1.4. Este paso genera como resultado un conjunto de documentos (para mayor detalle, revisar la sección 2.1.2.1).
- **Consolidar Resultados:** En este paso se identifican los documentos que se encuentran repetidos dentro de los resultantes, para finalmente generar un listado de documentos únicos (para mayor detalle, revisar la sección 2.1.2.2).
- **Primera Evaluación:** Este paso consiste en evaluar el título y el resumen de los documentos únicos, de acuerdo a los criterios de evaluación, definidos en la sección 2.1.1.5 (para mayor detalle, revisar la sección 2.1.2.3).
- **Primer Filtrado de Documentos:** En este paso se filtran los documentos únicos, para seleccionar los más importantes mediante el uso de las categorías (*Importante*, *Amplio* o *General*), definidas en la sección 2.1.1.6. Es decir, se seleccionan todos los documentos que cumplan con algunas de las categorías *Importante*, *Amplia* o *General* (para mayor detalle, revisar la sección 2.1.2.4).
- **¿Documentos Cumplen con Algún Criterio (PFD<sup>4</sup>)?:** En este paso se seleccionan los documentos que cumplen con algún criterio de las categorías de filtrado, y los que no cumplen son descartados.
- **Segunda Evaluación:** En este paso se evalúa todo el contenido de cada uno de los documentos, seleccionados en el paso anterior (para mayor detalle, revisar la sección 2.1.2.5).
- **Segundo Filtrado de Documentos:** En este paso se filtran los documentos seleccionados para elegir a los más importantes, mediante el uso de las categorías (*Importante*, *Amplio* o *General*), definidas en la sección 2.1.1.6 (para mayor detalle, revisar la sección 2.1.2.6).
- **¿Documentos Evaluados Cumplen con Algún Criterio (SFD<sup>5</sup>)?:** Los documentos evaluados por segunda vez, y que cumplen con algún criterio de las categorías de filtrado, finalmente se seleccionan, y aquellos que no cumplen se descartan (para mayor detalle, revisar la sección 2.1.3).

---

<sup>4</sup> Primer Filtrado de Documentos.

<sup>5</sup> Segundo Filtrado de Documento.



**Figura 2:** Proceso general de estudio sistemático de la literatura.

## 2.1.2. Ejecución del Estudio

En esta sección, se presenta la realización del estudio con los datos reales de la búsqueda.

### 2.1.2.1. Búsqueda de Documentos

Se realizó la búsqueda de los documentos en las diferentes bases de datos (definidas en la sección 2.1.1.2), de acuerdo a la cadena de búsqueda (definida en la sección 2.1.1.4), obteniendo los siguientes resultados (ver Tabla 6).

Bases de Datos	Documentos Encontrados
IEEE Xplore	96
ACM Digital Library	136
Springerlink	192
ScienceDirect	28
ISI Web of Knowledge	34
Scopus	224
<b>Total</b>	<b>710</b>

**Tabla 6:** Número de documentos encontrados en las bases de datos.

### 2.1.2.2. Consolidación de Resultados

La búsqueda de documentos en las diferentes bases de datos generó que un mismo documento fuera encontrado en más de una base de datos. Para resolver este inconveniente, se llevó a cabo el proceso de consolidación de los resultados (organizar y agrupar los documentos en una sola fuente), para identificar y descartar los documentos duplicados. Este proceso generó un total de 566 documentos únicos.

### 2.1.2.3. Primera Evaluación

Una vez consolidados los resultados de las diferentes bases de datos, se procedió a evaluar cada uno de los 566 documentos únicos. Este proceso consistió en evaluar el título y el resumen de dichos documentos, de acuerdo a los criterios de evaluación, definidos en la sección 2.1.1.5. En la Tabla 7, se presenta el resultado de la primera evaluación.

Áreas Evaluación	Modelos de Proceso	Enfoque Centrado en el Usuario	Enfoque Ágil	Despliegue de la AI
<i>Sin Relevancia (0)</i>	462	459	557	503
<i>Relevancia Baja (1)</i>	30	48	6	42
<i>Relevancia Media (2)</i>	41	35	3	21
<i>Relevancia Alta (3)</i>	33	24	0	0

**Tabla 7:** Resultado de la primera evaluación.

Como se puede observar en la Tabla 7, los documentos evaluados con una *Relevancia Alta* representan un porcentaje muy bajo en comparación con las otras categorías. De hecho, sólo el 5,82% de los documentos se evaluó con una *Relevancia Alta* en el área de *Modelos de Proceso*, y 4,23% en el área de *Enfoque Centrado en el Usuario*. Además,

no se encontraron documentos que fueran evaluados con una *Relevancia Alta* en las áreas de *Enfoque Centrado Ágil* y *Despliegue de la AI*.

Asimismo, en estos resultados no se encontraron documentos que fueran relevantes en todas las áreas de estudio. Por lo tanto, se podría concluir que no hay trabajos académicos que realicen una integración de la AI dentro de un proceso ágil de desarrollo centrado en el usuario. Sin embargo, la primera evaluación se realizó analizando sólo el título y el resumen de los 566 documentos. Es por esto que, para poder llegar a una conclusión más fehaciente respecto a los tipos de trabajos existentes en la revisión de la literatura, se efectuó un primer filtrado de estos documentos, para contar con un conjunto más reducido y así llevar a cabo una segunda evaluación para analizar en profundidad todo su contenido. Además, en esta segunda evaluación, se revisó en detalle la relevancia de los artículos resultantes de la primera selección.

#### 2.1.2.4. Primer Filtrado de Documentos

Para llevar a cabo la segunda evaluación, primero se procedió a filtrar los documentos más importantes. Este primer filtrado de documentos consistió en seleccionar los documentos, de acuerdo a las categorías de filtrados, definidos en la sección 2.1.1.6, generando como resultado un conjunto de 499 documentos descartados y 67 documentos seleccionados, siendo estos últimos los considerados para la segunda evaluación.

En la Tabla 8, se presenta el resultado de los 67 documentos seleccionados después de aplicar las categorías de filtrado. Como se puede observar en la Tabla 8, los documentos seleccionados corresponden a: 36 en la categoría *Importante*, 19 en la categoría *Amplio* y 12 en la categoría *General*. Un aspecto a destacar de estos resultados es que la mayoría de los documentos, el 57,62%, se seleccionaron principalmente por la categoría *Importante*.

Categorías para Filtrar Documentos	Documentos Seleccionados
<i>Importante</i>	36
<i>Amplio</i>	19
<i>General</i>	12
<b>Total</b>	<b>67</b>

**Tabla 8:** Documentos seleccionados después de la primera evaluación.

#### 2.1.2.5. Segunda Evaluación

Esta segunda evaluación consistió en evaluar el contenido completo de cada uno de los 67 documentos seleccionados (ver Tabla 8), aplicando los criterios de evaluación, definidos en la sección 2.1.1.5. En la Tabla 9, se presenta el resultado de esta segunda evaluación.

<b>Evaluación \ Áreas</b>	<b>Modelos de Proceso</b>	<b>Enfoque Centrado en el Usuario</b>	<b>Enfoque Ágil</b>	<b>Despliegue de la AI</b>
<i>Sin Relevancia (0)</i>	42	27	64	52
<i>Relevancia Baja (1)</i>	16	31	2	11
<i>Relevancia Media (2)</i>	3	8	1	4
<i>Relevancia Alta (3)</i>	6	1	0	0

**Tabla 9:** Resultado de la segunda evaluación.

Como se puede observar en la Tabla 9, de los 67 documentos seleccionados que se evaluaron por segunda vez, sólo un bajo porcentaje obtuvo una evaluación de *Relevancia Alta (3)*. Esta vez, el 8,95% de los documentos se evaluó con una *Relevancia Alta (3)* en el área de *Modelos de Proceso*, y el 1,49% en el área de *Enfoque Centrado en el Usuario*. Al igual que en la primera evaluación, tampoco se encontraron documentos que se consideraran relevantes en las áreas de *Enfoque Centrado Ágil* y *Despliegue de la AI*. Por lo tanto, el resultado de esta segunda evaluación permite responder la pregunta de investigación **PI<sub>1</sub>**, indicada en la sección 1.2.

- **PI<sub>1</sub>:** ¿Qué propuestas existen para integrar la Arquitectura de la Información dentro de un modelo de proceso ágil de desarrollo centrado en el usuario?

De acuerdo a los resultados de esta segunda evaluación, se puede afirmar de manera más fehaciente que no existen trabajos académicos que integren la AI a través de un marco ágil de desarrollo centrado en el usuario.

Por otro lado, los documentos seleccionados se pueden utilizar como fuente de información importante, para analizar en profundidad el estado actual de los modelos/metodologías de proceso para la AI. Es por esto que, se continuará analizando los documentos que resultaron seleccionados después de esta segunda evaluación.

#### **2.1.2.6. Segundo Filtrado de Documentos**

Una vez realizado el segundo filtrado de documentos, donde se evaluó todo el contenido de los 67 documentos seleccionados en el paso anterior, se eligieron nueve documentos, de acuerdo a los criterios de filtro, definidos en la sección 2.1.1.6. La Tabla 10 presenta el resultado de estos documentos.

Como se puede observar en la Tabla 10, los documentos seleccionados corresponden a: seis en la categoría *Importante*, dos en la categoría *Amplio* y uno en la categoría *General*. Un aspecto a destacar de estos resultados, es que nuevamente, al igual que en el primer filtrado, la mayoría de los documentos, el 66,66%, se seleccionaron principalmente por la categoría *Importante*.

Categorías para Filtrar Documentos	Documentos Seleccionados
<i>Importante</i>	6
<i>Amplio</i>	2
<i>General</i>	1
<b>Total</b>	<b>9</b>

**Tabla 10:** Documentos seleccionados después de la segunda evaluación.

A continuación, se presenta una descripción detallada de los resultados, obtenidos en este segundo filtrado de documentos.

### 2.1.3. Resultados Finales

La Tabla 11 se ha elaborado para presentar en detalle los nueve documentos seleccionados finalmente, y las diferentes evaluaciones que estos obtienen en las cuatro áreas de análisis.

Área 1: Modelos de Proceso	Área 2: Enfoque Centrado en el Usuario	Área 3: Enfoque Ágil	Área 4: Despliegue de la AI	Documentos Seleccionados Finalmente
2	1	1	1	Usability, Information Architecture and Semiotic Engineering Applied to the writing of support systems. (Rocio y Cerutti, 2016)
3	2	0	2	Information Architecture: For the Web and Beyond. (Rosenfeld et al., 2015)
3	1	0	2	10-Step Information Architecture on Wanfang Med Online. (Zhang et al., 2015)
2	1	1	1	Information Architecture Assessment of BPS Headquarter Official Website. (Nurcahyanti, 2014)
2	0	2	2	Pro iOS Web Design and Development: HTML5, CSS3, and JavaScript with Safari. (Picchi, 2011)

3	2	0	1	Guide to Creating Website Information Architecture and Content. (Sharlin et al., 2009)
3	3	0	2	LUCIA: Development of a Comprehensive Information Architecture Process Model for Websites. (Reichenauer, 2005)
3	2	0	1	Introduction to a User Interface Design/Information Architecture Process for Web Sites. (Lamar, 2001)
3	2	0	1	Evaluating Information Architecture: A Practical Guide to Assessing Web Site Organization. (Toub, 2000)

**Tabla 11:** Documentos seleccionados finalmente y su evaluación en las áreas de análisis.

Asimismo, se ha elaborado la Tabla 12, con el fin de visualizar de manera más clara el número de documentos que se seleccionaron finalmente, según el área de análisis y su relevancia.

<b>Áreas</b> <b>Evaluación</b>	<b>Modelos de Proceso</b>	<b>Enfoque Centrado en el Usuario</b>	<b>Enfoque Ágil</b>	<b>Despliegue de la AI</b>
<i>Sin Relevancia</i> (0)	0	1	6	0
<i>Relevancia Baja</i> (1)	0	3	2	5
<i>Relevancia Media</i> (2)	3	4	1	4
<i>Relevancia Alta</i> (3)	6	1	0	0

**Tabla 12:** Número de documentos seleccionados finalmente, según el área de análisis y su relevancia.

Como se puede observar en la Tabla 12, la mayoría de los documentos se seleccionaron con la categoría *Importante* y son provenientes, principalmente, del área de *Modelo De Proceso*. Además, esta área (*Modelo de Proceso*) concentra la mayor cantidad de documentos tipificados con una *Relevancia Alta* (3), seis en total. Por lo tanto, se puede indicar que la mayoría de los documentos seleccionados presentan modelos o metodologías para la AI.

También se puede destacar, que en el área de *Enfoque Centrado en el Usuario* se obtiene un documento evaluado con *Relevancia Alta* (3) y cuatro documentos con *Relevancia Media* (2), lo que demuestra la estrecha relación existente entre los modelos y metodologías para la AI con los enfoques centrados en el usuario.



Otro aspecto a destacar, es que no existen documentos evaluados con *Relevancia Alta* (3) en el área de *Enfoque Ágil* y que sólo un documento es evaluado con *Relevancia Media* (2), siendo la mayoría (seis documentos) evaluados *Sin Relevancia*. Esto demuestra la carencia de propuestas para el desarrollo ágil de la AI.

Asimismo, el principal aspecto a destacar, es que no se encontraron documentos que fueran relevantes en todas las áreas de análisis (*Modelos de Proceso*, *Enfoque Centrado en el Usuario*, *Enfoque Ágil* y *Despliegue de la AI*). No obstante, estos documentos se utilizarán para analizar en profundidad las actividades, las técnicas, los productos y los tipos de modelos que se consideran para integrar la AI.

Por otro lado, y dado el alto nivel práctico de la AI, también se realizaron búsquedas de artículos y libros en formatos y fuentes no académicos o científicos, sin conseguir resultados positivos. Por lo tanto, estos resultados permiten corroborar la hipótesis de partida **H1.1**, al concluir que no existen trabajos que realicen una integración de la AI dentro de un proceso ágil de desarrollo centrado en el usuario.

## 2.2. Arquitectura de la Información

Una vez realizado el estudio sistemático de la literatura, en esta sección se presentan los principales elementos conceptuales de la Arquitectura de la Información. Asimismo, se efectúa un análisis comparativo de los nueve documentos seleccionados en la sección anterior, con el fin de corroborar la hipótesis de partida **H1.2**: Las propuestas para integrar la Arquitectura de la Información actuales carecen de un modelo de proceso de desarrollo capaz de adaptarse a entornos ágiles y que considere a los usuarios finales durante todo el ciclo de vida del desarrollo de software.

De igual manera, con la realización de este estudio, también se busca responder a las siguientes preguntas de investigación (definidas anteriormente en la sección 1.2):

**PI<sub>2</sub>**: ¿Qué elementos de la Arquitectura de la Información cubren las propuestas metodológicas existentes?

**PI<sub>3</sub>**: ¿Qué propuestas existentes para la Arquitectura de la Información tienen una descripción completa y detallada de cada uno de los componentes presentados?

**PI<sub>4</sub>**: ¿En qué fases del ciclo de vida de la Arquitectura de la Información se centran las propuestas existentes?

**PI<sub>5</sub>**: ¿Qué propuestas existentes de la Arquitectura de la Información permiten diseñar sus productos centrados en las necesidades de los usuarios finales durante las diferentes etapas de desarrollo?

**PI<sub>6</sub>**: ¿Qué técnicas para integrar la Arquitectura de la Información permiten considerar las necesidades y capacidades de los usuarios finales, y adaptarse a entornos ágiles?

### 2.2.1. Definición

La AI es un paradigma creciente que se está incorporando paulatinamente en la mayoría de los proyectos software. La AI se define como la ciencia de la estructuración,

organización y gestión de la información, donde el arte del etiquetado, el hallazgo y la facilidad de uso son también importantes (Erlin et al., 2008). La AI cubre indistintamente aspectos, tales como el diseño de sitios Web, interfaces de dispositivos móviles, interfaces de máquinas dispensadoras, interfaces de juegos electrónicos, etc. Su principal objetivo es facilitar al máximo los procesos de comprensión y asimilación de la información, así como las tareas que ejecutan los usuarios en un espacio de información definido (Morville y Rosenfeld, 2006). La forma en que las personas interactúan con los entornos ricos en información digital está directamente influenciada por la AI (Elaine, 2002).

### 2.2.2. Arquitecto de la Información

El Arquitecto de la Información es la persona que realiza las tareas de descripción y estructuración de la información, con el objetivo de que los usuarios finales puedan recuperarla, manejarla y comprenderla de forma sencilla (Yusef y Francisco, 2003). El Arquitecto de la Información trabaja sobre las fases tempranas del software interactivo, principalmente en aplicaciones Web, tratando de encontrar un puente que una los conocimientos conceptuales que plantean los usuarios, dentro del espacio del problema, y la información de diseño que posteriormente necesitarán los analistas e ingenieros de la solución, que se encargarán de diseñar la interfaz Web final.

### 2.2.3. Entregables de la Arquitectura de la Información

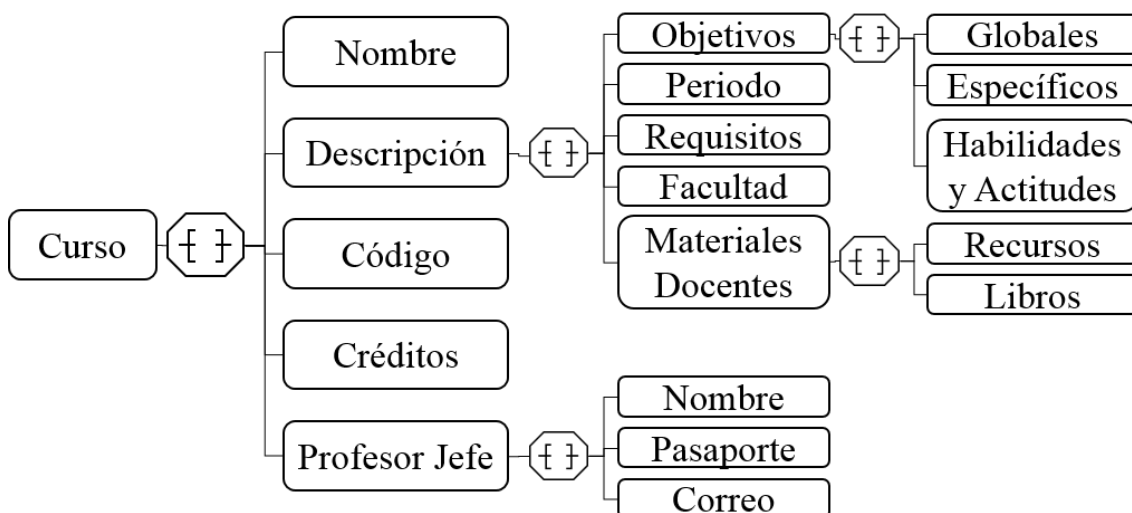
Los Arquitectos de la Información dependen de las representaciones visuales para comunicar lo que realmente hacen (Lima et al., 2015). Entre los entregables, utilizados por los Arquitectos de Información, están *los modelos de contenido*, *los planos (blueprints)*, *las maquetas (wireframes)*, *los inventarios de contenido* y *los vocabularios controlados* (Rosenfeld et al., 2015). Por un lado, *los modelos de contenido*, *los planos (blueprints)* y *las maquetas (wireframes)* permiten representar la estructura, el movimiento, el flujo y las relaciones entre el contenido. Por otro lado, *los inventarios de contenido* y *los vocabularios controlados* permiten identificar los contenidos existentes de un proyecto, y centrar la discusión en su contenido semántico. A continuación, se provee una descripción de todos estos entregables de la AI en los siguientes apartados.

#### a) Modelos de Contenido

Los modelos de contenido son *micro* arquitecturas de información, formadas por pequeños trozos de contenido, relacionados entre sí. Estos se basan en conjuntos coherentes de elementos y conexiones lógicas entre ellos. Un ejemplo de modelo de contenido puede ser la descripción de un producto de una tienda, donde sus elementos se corresponden a una lista de precios, la descripción y los datos de identificación del producto, que se utilizan para crear el modelo de contenido e identificar oportunidades para su reutilización, y controlar la granularidad de la información (Sachs, 2008).

La Figura 3 muestra un modelo de contenido de una universidad. Este modelo está compuesto por un elemento principal *Curso*, que a la vez está compuesto por cinco elementos descendientes (*Nombre*, *Descripción*, *Código*, *Créditos* y

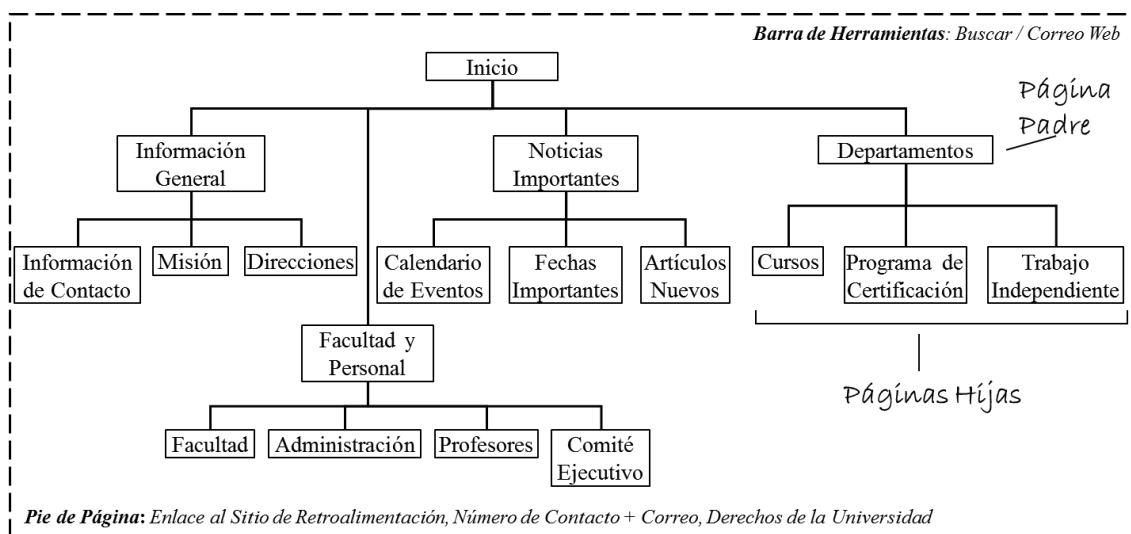
Profesor Jefe). Algunos de estos elementos descendientes del elemento principal (Curso) también contienen otros elementos descendientes para describirlos. Por ejemplo, el elemento secundario Profesor Jefe se describe con los tres elementos que descenden de él, es decir, Nombre, Pasaporte y Correo. Estos son los elementos que permiten indicar diferentes datos, relacionados con el profesor encargado del curso.



**Figura 3:** Modelo de contenido general.

#### a) Planos (Blueprints)

A diferencia de los modelos de contenido que se basan en las relaciones lógicas de los contenidos, los planos muestran las relaciones entre las páginas y otros componentes de datos. Los planos permiten representar las principales áreas de organización y rotulado de un producto (Morville y Rosenfeld, 2006). Además, permiten describir los aspectos estructurales de un producto y se representan con cajas relacionadas con flechas y texto.

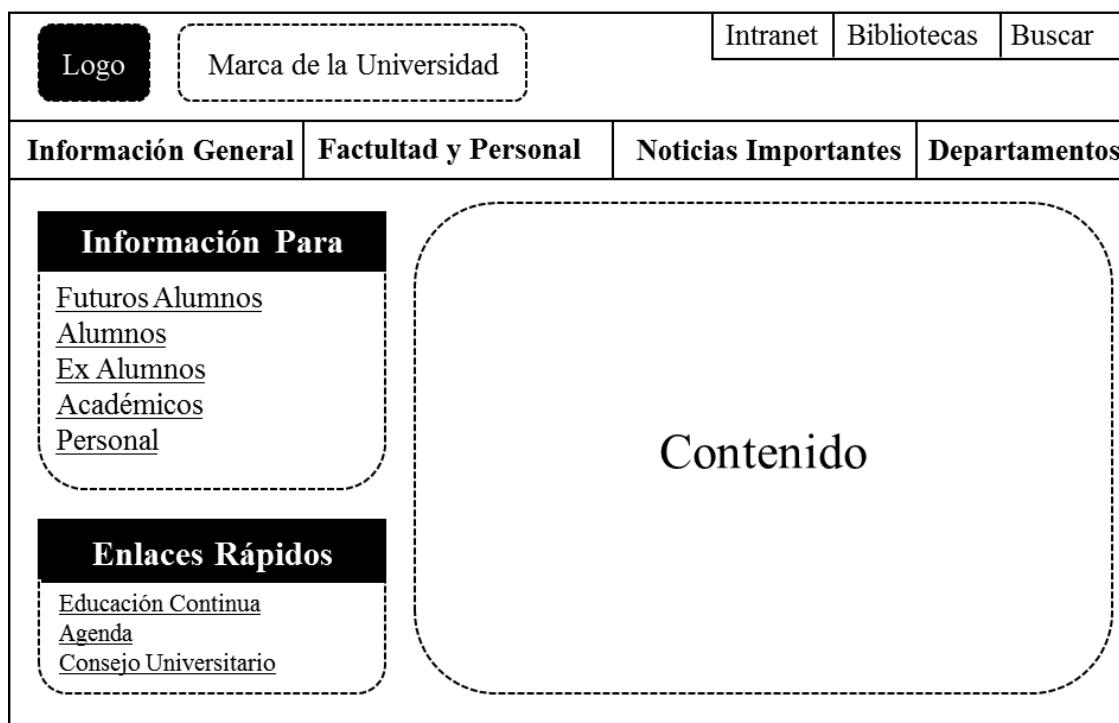


**Figura 4:** El plano de un sitio Web para una universidad.

La Figura 4 corresponde a un plano que muestra gráficamente cómo la información está organizada en un sitio de una universidad. Como se puede observar, Información General, Facultad y Personal, Noticias Importantes y Departamentos corresponden a los niveles principales del sitio Web. Cada nivel principal identifica, a la vez, las diferentes opciones que ofrece mediante sus niveles secundarios. Por ejemplo, el nivel principal Información General está compuesto por los niveles secundarios Información de Contacto, Misión y Direcciones que muestran que el nivel principal tendrá funcionalidades para obtener información sobre los contactos, las direcciones y la misión de la universidad. De igual manera, el mapa identifica la posición de la barra de herramientas y búsqueda del sitio Web.

### b) Maquetas (Wireframes)

A diferencia de los modelos de contenido y los planos que se basan en las relaciones de los contenidos y las páginas, las maquetas muestran el contenido de las páginas, identificando la estructura del contenido de un sitio Web. Éstas permiten al Arquitecto de la Información decidir qué grupos de componentes de contenido tienen prioridad, cómo agrupar los componentes de contenido y cómo ordenarlos.



**Figura 5:** Maqueta de la página principal de un sitio Web para una universidad.

Tal y como se puede observar en la Figura 5, el Arquitecto de la Información ha determinado que Información Para es de una prioridad más alta que los Enlaces Rápidos. Esta prioridad se pone de manifiesto por la posición superior de Información Para. Asimismo, se ha definido que Información General, Facultad y Personal, Noticias Importantes y Departamentos serán las opciones principales de navegación, y que el logo y la descripción de la marca

de la universidad se encontrarán en la parte superior izquierda del sitio Web. Cabe destacar que estas opciones principales de navegación se obtienen a partir de los niveles principales del sitio Web, identificados en el plano (ver Figura 4).

### c) Inventario de Contenidos

Los entregables anteriores (los modelos de contenido, los planos y las maquetas) se centran en representar las relaciones de los contenidos y de las páginas. En cambio, un inventario de contenidos identifica el contenido disponible y dónde éste se puede encontrar. Aquello permite conocer los contenidos que ofrece un sitio Web. La Figura 6 muestra una hoja de cálculo para un inventario de contenidos, creada para el sitio Web de una universidad. Como se puede observar, los contenidos se encuentran agrupados de acuerdo a los diferentes niveles de las páginas Web, identificados inicialmente en el plano (ver Figura 4), donde por cada contenido identificado se indica la dirección URL de su ubicación y las observaciones correspondientes.

ID	Nombre de la Página	Fuente del Contenido	Notas
0.0	<b>Página Principal</b>	<a href="http://www.universidadMG.com/">http://www.universidadMG.com/</a>	
1.0	<b>Facultades</b>	<a href="http://www.universidadMG.com/Facdes">http://www.universidadMG.com/Facdes</a>	
1.1	Academicos	<a href="http://www.universidadMG.com/Facdes/Acads">http://www.universidadMG.com/Facdes/Acads</a>	Enlaces a sitios web de los academicos
1.2	Personal	<a href="http://www.universidadMG.com/Facdes/Personal">http://www.universidadMG.com/Facdes/Personal</a>	
1.3	Ex-alumnos	<a href="http://www.universidadMG.com/Facdes/Ex">http://www.universidadMG.com/Facdes/Ex</a>	
2.0	Admisión	<a href="http://www.universidadMG.com/Facdes/Adm">http://www.universidadMG.com/Facdes/Adm</a>	
2.1	Carreras	<a href="http://www.universidadMG.com/Facdes/Carreras">http://www.universidadMG.com/Facdes/Carreras</a>	
2.2	Vías de Ingreso	<a href="http://www.universidadMG.com/Facdes/Ingreso">http://www.universidadMG.com/Facdes/Ingreso</a>	Norma 435
2.x	Información General	<a href="http://www.universidadMG.com/Facdes/Inform">http://www.universidadMG.com/Facdes/Inform</a>	
3.0	Campus	<a href="http://www.universidadMG.com/Facdes/Campus">http://www.universidadMG.com/Facdes/Campus</a>	
3.1	<b>Investigación</b>	<a href="http://www.universidadMG.com/Inv">http://www.universidadMG.com/Inv</a>	
3.2	Noticias	<a href="http://www.universidadMG.com/Inv/Noticias">http://www.universidadMG.com/Inv/Noticias</a>	
3.3	Descripción General	<a href="http://www.universidadMG.com/Inv/Descr">http://www.universidadMG.com/Inv/Descr</a>	Actualizado en mayo 2016
3.4	Centros de Investigación	<a href="http://www.universidadMG.com/Inv/Centros">http://www.universidadMG.com/Inv/Centros</a>	
3.5	Políticas de Incentivo	<a href="http://www.universidadMG.com/Inv/Políticas">http://www.universidadMG.com/Inv/Políticas</a>	
3.6	Convenios de Desempeño	<a href="http://www.universidadMG.com/Inv/Convenios">http://www.universidadMG.com/Inv/Convenios</a>	
4.0	<b>Bibliotecas</b>	<a href="http://www.universidadMG.com/Bibliotecas">http://www.universidadMG.com/Bibliotecas</a>	
5.0	<b>Contáctenos</b>	<a href="http://www.universidadMG.com/Contacto">http://www.universidadMG.com/Contacto</a>	Enlaces a direcciones de correo
6.0	<b>Política de Privacidad</b>	<a href="http://www.universidadMG.com/Privacidad">http://www.universidadMG.com/Privacidad</a>	
7.0	<b>Mapa del Sitio</b>	<a href="http://www.universidadMG.com/Mapa">http://www.universidadMG.com/Mapa</a>	Realizado con InterArch

**Figura 6:** Hoja de cálculo para inventario de contenidos de un sitio Web para una universidad.

### a) Vocabulario Controlado

A diferencia de los inventarios de contenido que se centran en la ubicación e identificación de los contenidos, un vocabulario controlado corresponde a una lista o índice de términos, que establecen relaciones unívocas y precisas entre ellos, así como con los conceptos representados (Hassan y Núñez, 2005). Los vocabularios controlados son matrices de metadatos que permiten la discusión sobre la priorización de los vocabularios.

La Figura 7 corresponde a un vocabulario controlado para una universidad. Como se puede observar, esta matriz contiene los diferentes vocabularios empleados en el sitio Web, de acuerdo a los contenidos identificados en el inventario de contenido (ver Figura

6). En esta matriz se realiza una descripción de cada uno de los términos empleados y se muestran ejemplos de sus posibles usos, además de una categorización del nivel de dificultad que representan en su mantenimiento. Por ejemplo, el vocabulario Tipo de Estudios se utiliza para identificar la oferta académica ofrecida por la universidad (Estudios de Grado, Másteres Oficiales y Programa de Doctorados), que tiene un nivel de dificultad Moderado para su mantenimiento.

Vocabulario	Descripción	Ejemplos	Mantenimiento
Admisión	Requisitos de admisión para los estudios	PAU; MG25; MG40; MG45; CFGS;	Difícil
Facultades	Centros docentes donde se imparten estudios especializados en alguna rama del saber	Ingeniería; Humanidades; Química y Biología; Economía; Ciencia; Ciencias Médicas; Arquitectura	Moderado
Tipo de Estudios	Tipos de ofertas académicas de la universidad	Estudios de Grado; Másteres Oficiales; Programa de Doctorados;	Moderado
Categoría de Becas	Clase de becas ofertadas para los alumnos	Ministerio de Educación; Gobierno Vasco; Beca de Excelencia; Ayudas de la Universidad	Fácil
Actividades Culturales	Actividades ofrecidas por la universidad para crear, difundir o desarrollar la cultura	Cursos de Verano; Música; Diálogos con la Cultura; Cerámica; Teatro; Sala de Exposiciones; Taller de Lectura	Fácil
Tecnologías para la Educación	Plataformas online para la educación	MOOCs; Moodle; Docencia en Red; Apoyo a la Docencia Digital	Moderado
Idioma	Nombre del lenguaje	Inglés; Francés; Español; Lituano	Fácil
Tipo de Documento	Propósito del objeto de contenido	Formulario; Instrucciones, Guía	Fácil
Créditos Académicos	Unidad de medida de la carga académica de las asignaturas	ECTS; Créditos	Fácil
Carreras	Oferta académica de la universidad	Grado en Biología; Grado en Física; Grado en Derecho; Grado en Ingeniería Informática	Moderado

**Figura 7:** Matriz de metadatos para una empresa de tecnología.

#### 2.2.4. Metodologías para la Arquitectura de la Información

Una vez presentados el término AI y los diferentes entregables que el Arquitecto de la Información debe elaborar, en esta sección se realizan una descripción y un análisis comparativo de las diferentes metodologías para la AI, con el objetivo de evaluar sus capacidades tanto para involucrar y considerar a los usuarios finales, como para adaptarse a los cambios de las organizaciones.

En concreto, se han recopilado nueve propuestas sobre la AI, obtenidas como resultado del estudio sistemático de la literatura, realizado en la sección 2.1 (ver Tabla 11). Estas propuestas son suficientemente representativas para abordar las metodologías de la AI más actuales y relevantes.

Por otro lado, se ha definido un conjunto de criterios para realizar la comparación de las propuestas de la AI. Dichos criterios permiten tener un marco común para analizar las características de estas propuestas y verificar si permiten integrar la AI dirigida por las expectativas de los usuarios, considerando los cambios de los requisitos. A continuación, se describen los criterios que se van a utilizar:

- **Centrado en el Usuario:** Este criterio se utilizará para identificar las propuestas que diseñan sus productos centrados en las necesidades de los usuarios finales,

durante las diferentes etapas de desarrollo (Granollers, 2004). Este criterio tomará los siguientes valores:

- *Si*: En los casos donde las propuestas involucren a los usuarios finales durante todo el modelo de proceso de desarrollo.
- *Parcialmente*: Para aquellas propuestas que involucren al usuario solamente en algunas de sus etapas.
- *No*: En los casos donde las propuestas no involucren a los usuarios finales.
- **Elementos Cubiertos de la AI**: Este criterio se utilizará para identificar las propuestas que desarrollan eficientemente la AI, cubriendo sus diferentes elementos. Actualmente, no existe un consenso total en cuanto a su definición única. Por el contrario, existen varias definiciones que intentan, en su conjunto, cubrir la amplitud de lo que conlleva especificar claramente lo que es la AI. No obstante, es posible identificar sus diferentes elementos, que están involucrados durante el desarrollo de un producto. Erlin et al. (Erlin et al., 2008) realizaron un estudio sobre la reseña y evolución de la AI en la era de la información, que permite identificar los diferentes elementos involucrados en su despliegue. Estos elementos corresponden a: *Navegación, Organización, Etiquetado, Búsqueda y Recuperación, y Presentación de Información*. Por lo tanto, estos elementos se utilizarán para revisar si son cubiertos por las propuestas.
- **Nivel de Descripción**: Este criterio se utilizará para identificar el nivel de detalle con el que las propuestas describen a los elementos para desplegar la AI. Por ejemplo, en algunas de las propuestas se realiza una descripción acabada y pormenorizada de cada una de las actividades, roles, técnicas, productos de entrada y de salida, y las relaciones existentes en la propuesta, facilitando la tarea de llevarlas a cabo. En cambio, en otras propuestas, el nivel de descripción es bajo, sin entregar los detalles suficientes de cada uno de los elementos presentados, ni tampoco se especifica la totalidad de los elementos para desplegar la AI. Por lo tanto, este criterio tomará los siguientes valores:
  - *Bajo*: Para los casos donde las propuestas se describen en términos generales, sin entregar los detalles suficientes para desplegar la AI.
  - *Medio*: Para aquellas propuestas que están descritas de forma detallada, pero que no están compuestas por todos los elementos suficientes para desplegar la AI. Asimismo, para aquellas propuestas que exhiben todos los elementos que son requeridos para desplegar la AI, pero no entregan el suficiente detalle de cada uno de ellos.
  - *Alto*: Para aquellas propuestas que describen cada uno de los elementos propuestos con el detalle suficiente, y que presentan una completa propuesta en cuanto a los diferentes elementos para desplegar la AI.
- **Alcance de la Propuesta**: Este criterio se utilizará para identificar las fases del ciclo de vida de la AI que las propuestas logran cubrir. Por ejemplo, algunas propuestas sólo se concentran en proveer diseños de los entregables de la AI, pero sin indicar el análisis necesario para elaborarlos o las actividades requeridas

para su mantención y evaluación. De este modo, el criterio tomará los siguientes valores:

- *Análisis*: Es donde se identifican los componentes principales e iniciales del proyecto y se investiga sobre el problema que se intenta resolver.
- *Diseño*: Es donde se lleva a cabo el trabajo para representar la AI del sitio Web. Esta fase tiene como objetivo diseñar el sistema de la AI y crear sus prototipos.
- *Implementación*: Es el conjunto de acciones para integrar, en el desarrollo final del sitio Web, la propuesta de la AI resultante de la fase de diseño.
- *Gestión*: Es el conjunto de acciones para mantener y garantizar en el tiempo la AI, mediante una evaluación y gestión continua. Esto incluye tareas frecuentes de seguimiento, respecto al desempeño del sitio Web y la captura de observaciones y comentarios por parte de los usuarios finales, para obtener oportunidades de mejora.
- *Evaluación*: Consiste en el uso de diferentes métodos y técnicas, orientadas a evaluar los diferentes aspectos de la AI.

Las fases que son cubiertas de manera incompleta, sin entregar suficiente detalle, serán señaladas con la etiqueta (-). Por ejemplo: la etiqueta “*Evaluación (-)*” indica que la propuesta presenta una etapa de evaluación dentro de su proceso para la AI, pero no entrega los detalles suficientes para llevarla a cabo.

- **Flexibilidad y adaptabilidad**: Este criterio se utilizará para identificar la idoneidad de las propuestas, para implementar los requisitos de manera iterativa e incremental, así como sus habilidades para responder rápidamente a las exigencias y a los cambios del entorno de las organizaciones. Por ejemplo, se analizarán las propuestas para conocer si éstas requieren de una extensa planificación o tienen un modelo de proceso de desarrollo (por ejemplo, cascada), que demanda numerosos controles y políticas que dificultan su adaptación a entornos cambiantes.

### 2.2.5. Descripción de las Metodologías para la Arquitectura de la Información

Una vez descritos los criterios escogidos para analizar las fuentes sobre la AI, en la Tabla 13 se presenta la descripción de las propuestas para la AI, así como sus fortalezas y debilidades, de acuerdo a los criterios definidos. Más específicamente, en esta tabla se identifican y se detallan las diferentes actividades y fases propuestas en las metodologías para la AI. Asimismo, estas propuestas se analizan para verificar sus capacidades para considerar e integrar a los usuarios finales y, a la vez, proveer una respuesta eficiente en los entornos cambiantes donde operan las organizaciones.



Met. para la AI	Descripción	Fortalezas	Debilidades
(Rocio y Cerutti, 2016)	<p>En este trabajo se estudia la eficiencia de varias directrices de usabilidad, entre ellas la AI, aplicada a los materiales educativos en interfaces virtuales. Asimismo, se presenta una secuencia de seis pasos para integrar la AI.</p> <p>El primer paso, <i>Definición de los Objetivos</i>, consiste en definir los objetivos relacionados con las prioridades tanto del proyecto como del contenido. En el segundo paso, <i>Estudio del Público Objetivo</i>, se debe identificar y estudiar las características y rasgos del grupo de personas que requiere el sitio Web. El tercer paso, <i>Creación de Escenarios</i>, consiste en describir los diferentes escenarios de uso, relacionados con el sitio Web, con el fin de estudiar los problemas críticos del proyecto y prever sus posibles soluciones. En el cuarto paso, <i>Análisis de Competidores</i>, se analizan las ventajas competitivas, recursos, capacidades, fortalezas y debilidades de los actuales y potenciales productos, ofrecidos por los competidores relacionados con el sitio Web. El quinto paso, <i>Información</i>, trata de reunir la información relacionada con los competidores, que se relaciona con el paso anterior. Finalmente, el último paso, <i>Catalogación de la Información</i>, consiste en catalogar la información para generar un prototipo relacionado con el contenido del proyecto.</p>	<p>La única fortaleza que se puede destacar de este trabajo es que propone incorporar aspectos de la usabilidad, relacionados con la presentación y la organización de la AI.</p>	<p>La principal debilidad de este trabajo es que no provee suficiente información respecto a los pasos propuestos para integrar la AI. Además, no se concreta ninguno de los elementos propuestos. Esta propuesta sólo se centra en proveer algunos ejemplos de los resultados de los pasos. Finalmente, aun cuando se plantea concretar aspectos relacionados con la usabilidad, en los pasos propuestos no se especifican ni identifican oportunidades de interacción con los usuarios finales.</p>

<p>(Rosenfeld et al., 2015)</p>	<p>En este trabajo se describe una metodología para integrar la AI en entornos Web, compuesta por cinco fases: <i>Investigación, Estrategia, Diseño, Implementación y Gestión</i>.</p> <p>La primera fase, <i>Investigación</i>, tiene como objetivo obtener una comprensión de alto nivel de los objetivos, el contexto empresarial, la AI existente, el contenido y las audiencias previstas. Aquí se recomienda centrar el proceso de la investigación en el contexto, el contenido y los usuarios.</p> <p>La siguiente fase, <i>Estrategia</i>, es el puente entre la <i>Investigación</i> y el <i>Diseño</i> de la AI. Esta fase tiene como objetivo elaborar un informe sobre el plan de estrategia para la AI.</p> <p>En la fase de <i>Diseño</i>, se forma una estrategia de alto nivel, donde se crean planos detallados, maquetas y esquemas de metadatos, que serán utilizados por los diseñadores gráficos, Analistas/Ingenieros del Software, autores de contenido y el equipo de producción. En esta fase, se describen diferentes tipos de entregables que el Arquitecto de la Información debe generar y se identifican las tareas más importantes que se deben llevar a cabo.</p> <p>En la fase de <i>Implementación</i>, se pone a prueba lo que está construido para el sitio.</p> <p>Finalmente, en la fase de <i>Gestión</i>, se propone efectuar una evaluación continua de la AI, que incluye tareas diarias para etiquetar y gestionar los documentos.</p>	<p>Una de las principales fortalezas de esta metodología es la propuesta de un conjunto variado de herramientas y métodos para <i>el Análisis y el Diseño</i> de la AI. Esto permite cubrir sus diferentes elementos (<i>Navegación, Organización, Etiquetado, Búsqueda y Recuperación y Presentación de Información</i>). Además, se identifican diferentes tareas que el Arquitecto de la Información debe llevar a cabo, así como los resultados esperados (entregables).</p> <p>Asimismo, se puede destacar que, en la fase de <i>Diseño</i>, se identifican y describen diferentes tipos de entregables que el Arquitecto de la Información debe generar como resultado de las fases de <i>Investigación y Estrategia</i>. Esto permite vislumbrar algunas de las tareas más importantes que el Arquitecto de la Información debe llevar a cabo.</p>	<p>Una de las debilidades de esta propuesta es que no presenta una descripción detallada de las fases de <i>Implementación y Gestión</i>. Además, las técnicas y los métodos recomendados están sólo enfocados en las etapas iniciales del proyecto (<i>Investigación, Estrategia y Diseño</i>), y no existen recomendaciones para evaluar la AI. Finalmente, la propuesta presenta un ciclo de vida en cascada (secuencial-lineal), en el que no se iteran las diferentes fases del modelo de proceso, donde los usuarios sólo son considerados en etapas iniciales.</p>
---------------------------------	--	---	---

<p>(Zhang et al., 2015)</p>	<p>En este trabajo, se propone un proceso para integrar la AI, enfocado principalmente en los servicios de información médica. El proceso está compuesto por 10 pasos secuenciales:</p> <p>El primer paso, <i>Investigación de los Usuarios</i>, consiste en identificar las diferentes características de los usuarios (tareas, necesidades de información, comportamiento de búsqueda, entre otras). El segundo paso, <i>Análisis del Contenido</i>, consiste en describir el contenido en base a diferentes categorías (tipo de contenido, objeto de contenido, cantidad y estructura de los datos). El tercer paso, <i>Análisis del Contexto</i>, consiste en analizar los servicios ofrecidos de acuerdo a diferentes aspectos (técnicos, políticos, culturales, entre otros). El cuarto paso, <i>Construcción de un Marco de Metadatos</i>, trata de desarrollar un marco para gestionar las fuentes bibliográficas, basado en cuatro elementos (autor, institución, descriptor y financiación). El quinto paso, <i>Sistema de Clasificación</i>, consiste en catalogar a los usuarios, de acuerdo a diferentes sistemas de clasificación. Posteriormente, el sexto paso, <i>Aplicación de Tesauros</i>, consiste en generar una lista de palabras y términos utilizados en el proyecto. El séptimo paso, <i>Definición de la Estrategia de la AI</i>, trata de crear una estrategia de la AI, para guiar el estilo y las características del sitio Web. El octavo paso, <i>Gestión del Proyecto</i>, consiste en generar un plan para gestionar el proyecto y la estrategia de la AI. Asimismo, el noveno paso, <i>Desarrollo de Planos</i>, consiste en construir los planos del proyecto con los diferentes niveles de la información. El último paso, <i>Interfaz del Usuario</i>, trata de construir la interfaz del proyecto de acuerdo a los requisitos de los usuarios.</p>	<p>La única fortaleza de esta propuesta es el hecho de proveer ejemplos concretos de los resultados de los diferentes pasos propuestos para integrar la AI.</p>	<p>La principal debilidad de esta propuesta es la escasa presentación de información, respecto a la descripción de los pasos propuestos para integrar la AI. Tampoco se identifican iteraciones entre los pasos, ni se especifican las entradas y las salidas requeridas. Además, los usuarios sólo son considerados en el <i>Análisis</i> de la AI.</p>
-----------------------------	---	---	--

(Nurcahyanti, 2014)	<p>En esta fuente, se propone integrar la AI en base a una metodología compuesta por tres fases: <i>Investigación</i>, <i>Diseño</i> y <i>Evaluación</i>.</p> <p>Mientras que la fase de <i>Investigación</i> consiste en entender el contexto, el contenido y los usuarios del sitio Web, la fase de <i>Diseño</i> consiste en realizar una organización del contenido, en base a una clasificación de tarjetas, y crear entregables y prototipos del sitio Web. La última fase, <i>Evaluación</i>, consiste en realizar pruebas cualitativas y cuantitativas a los usuarios.</p>	<p>La única fortaleza de esta fuente es que provee recomendaciones para analizar el contexto y el contenido de los sitios Web.</p>	<p>La principal debilidad de esta fuente es el hecho de no presentar suficiente información de las fases propuestas para integrar la AI. En esta fuente sólo se entrega un exiguo detalle de la fase inicial de <i>Investigación</i>, centrándose únicamente en el análisis del contexto y contenido. Por otro lado, tampoco se proveen ejemplos de resultados de las fases propuestas, con el fin de distinguir los objetivos perseguidos.</p>
---------------------	--	--	---

<p>(Picchi, 2011)</p>	<p>En esta fuente se presenta una propuesta para la AI, compuesta por tres fases principales: <i>Descubrimiento</i>, <i>Análisis</i> y <i>Arquitectura</i>.</p> <p>La fase de <i>Descubrimiento</i> consiste en coleccionar la información acerca del contexto de mercado del sitio Web. Esta fase contiene una actividad relacionada con la investigación de la información, con el fin de iniciar la recogida de información, y utilizarla como fuente para las decisiones de las fases posteriores. Además, se proveen recomendaciones específicas sobre técnicas para analizar el contexto del sitio Web.</p> <p>En la fase <i>Análisis</i>, se comienza a trabajar con la información recogida en la fase anterior (<i>Descubrimiento</i>), con el fin de desarrollar una estrategia de la AI. Esta fase está compuesta por una actividad para gestionar la información, que tiene como objetivo planificar el enfoque general de la organización, que se llevará a cabo para la gestión de la información. Asimismo, en esta actividad, se presentan diversas recomendaciones para gestionar la información.</p> <p>Es en la última fase de <i>Arquitectura</i>, donde se lleva a cabo el trabajo de diseño. Esta fase está compuesta por siete actividades: <i>Arquitectura de Contenidos</i>, <i>Diseño de Experiencia</i>, <i>Experiencia de Usuario</i>, <i>Diseño de Información</i>, <i>Ingeniería de Usabilidad</i>, <i>Diseño de Interacción</i> e <i>Interacción Persona-Ordenar</i>.</p> <p>En la actividad de la <i>Arquitectura de Contenido</i>, se diseña, a un alto nivel, el contenido que comprenderá el sitio Web y se intenta identificar cómo este contenido será estructurado y</p>	<p>Una de las fortalezas de esta propuesta es que permite analizar y diseñar AI, cubriendo los aspectos relacionados con la <i>Navegación</i>, la <i>Organización</i> y la <i>Presentación de la Información</i>. Se propone el uso de un conjunto variado de técnicas, para utilizarlas en cada una de las fases. Además, en esta propuesta se especifican diversos documentos de salida que se generan en cada una de las fases descritas.</p>	<p>Una de las debilidades de esta propuesta es que no hay claridad respecto a la interrelación entre las diferentes fases y las actividades planteadas. Además, no se describen con suficiente detalle las diferentes técnicas propuestas, ni tampoco se especifica la secuencia de ejecución de las diferentes fases propuestas. Aun cuando se provee un conjunto variado de recomendaciones, no se identifican técnicas para evaluar la AI, centrándose sólo en su <i>Análisis</i> y <i>el Diseño</i>. Tampoco se especifican actividades ni tareas para cubrir aspectos de la AI, relacionados con <i>el Etiquetado</i>, <i>la Búsqueda</i> y <i>la Recuperación de la Información</i>.</p>
-----------------------	--	--	--

	<p>organizado. Similarmente, en la actividad de <i>Diseño de Experiencia</i>, se obtiene una visión más amplia del diseño y se considera la experiencia total que el usuario va a tener. Posteriormente, en la actividad de <i>Experiencia de Usuario</i>, se diseña la forma en que el contenido del sitio Web está organizado, considerando los diferentes contextos de los usuarios. Además, en la actividad de <i>Diseño de Información</i>, se identifica cómo se transmite la información, ya sea en términos de texto o de gráficos. La siguiente actividad sobre la <i>Ingeniería de Usabilidad</i> se centra en elaborar patrones de diseño. De un modo similar, en la actividad de <i>Diseño de Interacción</i>, se inicia el diseño de la interfaz de usuario que permite al usuario interactuar con el sitio Web. Finalmente, en la actividad sobre la <i>Interacción Persona-Ordenador</i> se describen los aspectos relacionados con los usuarios.</p>		
(Sharlin et al., 2009)	<p>En esta propuesta se describe una guía para la creación del contenido y de la AI, basada en ocho fases. Esta propuesta considera las expectativas, las necesidades y los intereses de los visitantes de un sitio Web. Además, en algunas de las fases, se realizan recomendaciones específicas sobre herramientas y productos que se deberían generar.</p> <p>La fase inicial consiste en definir las metas principales de las partes interesadas. En la segunda fase, se identifican los objetivos y las expectativas de los usuarios, a los que tiene que dar solución el sitio Web a desarrollar. En la tercera fase, se definen las áreas de contenido del sitio Web, analizando el contenido que se tiene y las piezas que hay que añadir, modificar o descartar del nuevo sitio Web. Posteriormente, en la cuarta fase, se agrupan y etiquetan las áreas de contenido,</p>	<p>La principal fortaleza de esta propuesta es que permite diseñar y analizar la AI considerando las preferencias de las partes interesadas y los usuarios finales. Además, esta propuesta permite cubrir los aspectos de la AI, relacionados con <i>la Navegación, la Organización y el Etiquetado</i>.</p>	<p>La principal debilidad de esta propuesta es que presenta escasa información respecto a las diferentes fases y elementos para la AI. Además, esta propuesta sólo se concentra en su <i>Análisis y el Diseño</i> y no se especifican actividades ni tareas para cubrir aspectos relacionados con <i>la Búsqueda, la Recuperación y la Presentación de la Información</i>.</p>

	con el fin de que la navegación sea más intuitiva para los usuarios. Mientras que en la quinta fase se crea una representación visual de las áreas de contenido, en la sexta fase se establece un esquema de la estructura de navegación del sitio Web. En la séptima fase, se deben etiquetar las áreas de contenido y, finalmente, en la última fase, se crea un modelo para representar cómo las áreas de las páginas se organizarán.		
(Reichenauer, 2005)	<p>En esta fuente se propone una metodología para la AI, compuesta por siete fases: <i>Descubrimiento, Análisis, Diseño, Evaluación, Revisión y Documentación, Implementación y Mantenimiento</i>. Cada fase se compone de una gran cantidad de procesos (que describen las entradas, las salidas, los responsables y los métodos) y diferentes tipos de relaciones existentes entre estos.</p> <p>En la fase <i>Análisis</i>, se examinan la situación actual del sitio Web, los competidores, los usuarios finales, el contexto de uso y los requisitos de los proveedores de contenido.</p> <p>La fase de <i>Diseño</i> tiene como objetivo diseñar el sistema de la AI, desarrollar la marca para el sitio Web y crear prototipos.</p> <p>La fase de <i>Revisión y Documentación</i> trata de revisar el prototipo del sitio, así como revisar y documentar la guía de desarrollo de contenidos y el sistema de la AI.</p> <p>La fase de <i>Mantenimiento</i> tiene como objetivo la manutención del contenido, de los problemas técnicos, del sistema de la AI y de la gestión de la marca.</p>	<p>Una de las principales fortalezas de esta metodología es que permite desplegar todos los aspectos de la AI (<i>Navegación, Organización, Etiquetado, Búsqueda y Recuperación, y Presentación De Información</i>). Además, esta fuente presenta una variada propuesta de métodos para evaluar e integrar la AI, considerando a los usuarios finales en las diferentes fases. Asimismo, la propuesta realiza una descripción completa de una gran cantidad de fases, necesarias para integrar la AI.</p>	<p>Una de las debilidades detectadas en esta propuesta es que se requiere de una extensa planificación para implementarla, demandando numerosos controles y políticas. Esto dificulta la adaptación a entornos cambiantes y la habilidad a responder de manera rápida.</p>

(Lamar, 2001)	<p>En esta fuente se presenta un proceso de seis pasos para la AI, basado en el entendimiento de los objetivos de la compañía y de los usuarios.</p> <p>El primer paso, <i>Entender los Objetivos de la Compañía</i>, tiene como objetivo identificar y analizar los objetivos del proyecto y los fines hacia los que está orientada la actividad de la organización. El segundo paso, <i>Definir la Audiencia</i>, consiste en entender y determinar las tareas y objetivos de las audiencias hacia las que se orientará el sitio Web, mientras que el tercer paso, <i>Definir la Información y la Necesidad de Interactividad</i>, consiste en identificar las necesidades de información de los usuarios y las prioridades de los contenidos, con el fin de determinar los requisitos de interacción de los usuarios con los contenidos del sitio Web. El cuarto paso, <i>Crear un Diagrama del Sitio</i>, trata de definir los elementos de contenido del sitio Web que cubren las necesidades, identificadas en los pasos anteriores. El quinto paso, <i>Crear una Maqueta</i>, consiste en identificar el sistema de navegación del sitio Web y los elementos de las páginas que se deben destacar. Finalmente, el sexto paso, <i>Consultar sobre la Producción del Sitio</i>, consiste en medir el grado de satisfacción de los usuarios con respecto al sitio Web.</p>	<p>Una de las principales fortalezas de esta propuesta es que el proceso permite identificar diferentes oportunidades para interactuar con los usuarios y obtener información de ellos. Asimismo, se identifican algunas interacciones entre las salidas y los entregables, generadas por los pasos. Además, la propuesta permite <i>el Análisis y el Diseño</i> de la AI, cubriendo aspectos relacionados con <i>la Navegación, la Organización y el Etiquetado</i>.</p>	<p>La principal debilidad de esta propuesta es que contiene una descripción en términos generales, sin entregar detalles suficientes ni la totalidad de los elementos que componen un proceso para la AI. Esta propuesta efectúa algunas recomendaciones respecto a la necesidad de evaluar los diseños de la AI, no obstante, no se especifican métodos y/o técnicas para llevarlas a cabo. Tampoco se especifican fases para su <i>Gestión y la Implementación</i>. Finalmente, aun cuando se identifican oportunidades para interactuar con los usuarios, éstas sólo se concentran en <i>el Análisis y el Diseño</i> de la AI.</p>
---------------	--	---	---



(Toub, 2000)	<p>En esta fuente se propone una guía para evaluar la AI, identificando sus diferentes aspectos que han de evaluarse. Estos aspectos son agrupados en tres tipos de evaluaciones: <i>la Evaluación de la Estructura</i>, <i>la Evaluación de la Agrupación</i> y <i>la Evaluación Del Etiquetado</i>.</p> <p>La <i>Evaluación de la Estructura</i> se refiere a la integridad (es decir, la inclusión o exclusión de contenido y de funcionalidad) y la forma (por ejemplo, la jerarquía vs bases de datos) del sitio en su conjunto.</p> <p>La <i>Evaluación de la Agrupación</i> se refiere a la coherencia y la integridad de las categorías de objetos de contenido.</p> <p>Finalmente, la <i>Evaluación del Etiquetado</i> se refiere al nombre o el icono de un objeto de contenido, como el título de una página o el título de una categoría o encabezamiento. En los tres tipos de evaluación, se identifican y se detallan los diferentes aspectos a evaluar, indicando recomendaciones específicas de las técnicas de evaluaciones que se deben utilizar.</p>	<p>En esta propuesta, junto con especificar diferentes técnicas para evaluar la AI, también se identifican qué aspectos se deben evaluar. Esto permite tener mayor claridad respecto a cómo modificar las técnicas de evaluación de otras áreas para adaptar su uso en la AI. También se destaca el hecho de que se logra identificar diferentes tipos de evaluaciones que deben llevarse a cabo de acuerdo a la fase en que se encuentre el proyecto. Además, se consideran evaluaciones concentradas y dirigidas por los usuarios finales.</p>	<p>Una de las debilidades detectadas en esta propuesta es que no considera evaluaciones para los aspectos de la AI, relacionados con <i>la Búsqueda</i>, <i>la Recuperación</i> y <i>la Presentación de la Información</i>. Además, aun cuando esta propuesta está descrita de forma detallada, no está compuesta por todos los elementos suficientes que componen un proceso para la AI, centrándose sólo en <i>la Evaluación</i>. Finalmente, la propuesta requiere de una extensa disposición para llevar a cabo las evaluaciones, y demanda numerosos controles para implementarlas.</p>
--------------	--	--	--

**Tabla 13:** Descripción de las metodologías para la Arquitectura de la Información.

### 2.2.6. Análisis Comparativo de las Metodologías para la Arquitectura de la Información

Como resumen de la revisión de las propuestas de la AI, descritas en la sección anterior, en la Tabla 14 se detallan las principales características de las metodologías para la AI, de acuerdo a los criterios definidos en la sección 2.2.4.

Met. para la AI	Centrado en el Usuario	Elementos Cubiertos de la AI	Nivel de Descripción	Alcance de la Propuesta	Flexibilidad y Adaptabilidad
(Rocio y Cerutti, 2016)	No	-Organización	Bajo	-Análisis (-) -Diseño (-)	No
(Rosenfeld et al., 2015)	Parcialmente	-Navegación -Organización -Etiquetado -Búsqueda y recuperación -Presentación de información	Alto	-Análisis -Diseño -Implementación (-) -Gestión (-)	No
(Zhang et al., 2015)	Parcialmente	-Organización -Presentación de información	Bajo	-Análisis (-) -Diseño (-)	No
(Nurcahyanti, 2014)	No	-Organización -Presentación de información	Bajo	-Análisis -Diseño (-) -Evaluación (-)	No
(Picchi, 2011)	Parcialmente	-Navegación -Organización -Presentación de información	Bajo	-Análisis -Diseño	No
(Sharlin et al., 2009)	Parcialmente	-Navegación -Organización -Etiquetado	Bajo	-Análisis -Diseño	No
(Reichner, 2005)	Si	-Navegación -Organización -Etiquetado -Búsqueda y recuperación -Presentación	Alto	-Análisis -Diseño -Evaluación -Implementación - Gestión	No

		de información			
(Lamar, 2001)	Parcialmente	-Navegación -Organización -Etiquetado	Bajo	-Análisis -Diseño -Evaluación (-)	No
(Toub, 2000)	Parcialmente	-Organización -Etiquetado	Medio	-Evaluación	No

**Tabla 14:** Resumen del análisis comparativo de las fuentes de estudios seleccionadas.

Como se puede observar en la Tabla 14, la mayoría de las propuestas no cubren todas las fases del ciclo de vida de la AI, centrándose principalmente en las fases de análisis y diseño. De hecho, sólo las propuestas (Toub, 2000) y (Reichenauer, 2005) presentan recomendaciones de actividades y técnicas para la evaluación de la AI (Criterio *Alcance de la Propuesta: Evaluación*). De manera similar, las propuestas (Nurcahyanti, 2014 y (Lamar, 2001) también sugieren realizar su evaluación. No obstante, estas propuestas se describen superficialmente (-), sin proveer suficiente información para llevar tal evaluación a cabo.

Se puede observar también que la metodología (Reichenauer, 2005) es la única que proporciona un enfoque centrado en el usuario para integrar la AI (Criterio *Centrado en el Usuario: Si*). Por el contrario, en la mayoría de las propuestas este aspecto se realiza de manera parcial, (Rosenfeld et al., 2015; Zhang et al., 2015; Picchi, 2011; Sharlin et al., 2009; Lamar, 2001; Toub, 2000), es decir, los usuarios finales son considerados en las fases iniciales y no durante todo el modelo de proceso de desarrollo.

Asimismo, las metodologías (Rosenfeld et al., 2015) y (Reichenauer, 2005) son las únicas que tienen un adecuado nivel de descripción de las propuestas (Criterio *Nivel de Descripción: Alto*) y cubren todos los elementos de la AI (Criterio *Elementos Cubiertos de AI*) en las diferentes fases de su ciclo de vida (Criterio *Alcance de la Propuesta*).

En cuanto al criterio de *Flexibilidad y Adaptabilidad*, la metodología (Picchi, 2011) es la única que sugiere integrar la AI de manera ágil. No obstante, en esta propuesta no se identifican los detalles necesarios (modelo de proceso, actividades, tareas, relaciones, entre otras) para realizar una correcta integración de la AI en entornos cambiantes. Por lo tanto, es posible indicar que ninguna de las metodologías presenta flexibilidad y adaptabilidad para responder de una manera ágil y flexible en los entornos cambiantes, y requieren un esfuerzo considerable para ser adaptadas.

Finalmente, estos resultados permiten corroborar la hipótesis de partida **H1.2**, al concluir que las metodologías para la AI no presentan un modelo de proceso de desarrollo que sea centrado en el usuario y que, a la vez, pueda responder de una manera ágil y flexible a los entornos cambiantes.

Por otro lado, los inconvenientes anteriores se pueden abordar mediante la flexibilidad y adaptabilidad que ofrecen las metodologías ágiles que permiten responder rápidamente a los cambios del entorno. No obstante, las metodologías ágiles incluyen características

específicas, que en algunos casos dificultan el Diseño Centrado en Usuario, siendo éste un factor esencial para implementar una sólida AI.

Por lo tanto, se hace necesario analizar la viabilidad de las metodologías ágiles para integrar el Diseño Centrado en el Usuario, con el fin de discernir claramente cuáles son las metodologías ágiles más adecuadas para el desarrollo centrado en el usuario de la AI. Es por esto que, en la siguiente sección, se estudiarán las metodologías ágiles, después de este análisis comparativo, y se buscará corroborar la hipótesis de partida **H1.3**: La mayoría de las metodologías ágiles presentan dificultades para integrar el Diseño Centrado en el Usuario y sólo consideran a los usuarios finales de manera parcial durante el desarrollo de software.

Por último, estos resultados permiten responder las preguntas de la investigación **PI<sub>2</sub>**, **PI<sub>3</sub>**, **PI<sub>4</sub>** y **PI<sub>5</sub>**, definidas en la sección 1.2:

- **PI<sub>2</sub>**: ¿Qué elementos de la Arquitectura de la Información cubren las propuestas metodológicas existentes?

Las propuestas analizadas para la AI se centran principalmente en cubrir los aspectos relacionados con la organización y la navegación de la información, mientras que los elementos de búsqueda y recuperación son los menos considerados por las propuestas. Destacan las metodologías (Rosenfeld et al., 2015) y (Reichenauer, 2005), por ser las únicas que cubren todos los elementos de la AI (navegación, organización, etiquetado, búsqueda y recuperación, y presentación de información).

- **PI<sub>3</sub>**: ¿Qué propuestas existentes para la Arquitectura de la Información tienen una descripción completa y detallada de cada uno de los componentes presentados?

Las metodologías (Rosenfeld et al., 2015) y (Reichenauer, 2005) son las que presentan un adecuado nivel de descripción de sus elementos presentados. En la mayoría de los casos (6 de las 9 metodologías), las propuestas sólo se especifican en términos generales, sin proveer el suficiente detalle para llevarlas a cabo apropiadamente.

- **PI<sub>4</sub>**: ¿En qué fases del ciclo de vida de la Arquitectura de la Información se centran las propuestas existentes?

La mayoría de las propuestas se centran en las fases de análisis y diseño del ciclo de vida de la AI. Destaca la metodología (Reichenauer, 2005) por ser la única que cubre todo el ciclo de vida de la AI.

- **PI<sub>5</sub>**: ¿Qué propuestas existentes de la Arquitectura de la Información permiten diseñar sus productos centrados en las necesidades de los usuarios finales durante las diferentes etapas de desarrollo?

La metodología (Reichenauer, 2005) es la única que presenta un enfoque para involucrar a los usuarios finales. Destacan, entre las propuestas que involucran de manera parcial a los usuarios finales, los trabajos de (Rosenfeld et al., 2015) y (Toub, 2000) que proveen enfoques para involucrar a los usuarios finales con un nivel adecuado de descripción. No obstante, estas propuestas sólo se concentran

en algunas de las fases del ciclo de vida de la AI (Análisis, Diseño y Evaluación).

- **PI6:** ¿Qué técnicas para integrar la Arquitectura de la Información permiten considerar las necesidades y capacidades de los usuarios finales, y adaptarse a entornos ágiles?

En la Tabla 15, se presenta el conjunto de técnicas ágiles para integrar la AI, así como las fases recomendadas donde éstas se pueden aplicar: Análisis (A), Diseño (D), Evaluación (E), Implementación (I) y Gestión (G). Además, cada una de las metodologías para la AI asociada se detalla denotando su referencia bibliográfica correspondiente. Estas técnicas se han obtenido desde un segundo análisis de las nueve propuestas para la AI. Este segundo análisis se realizó para obtener un conjunto de actividades, técnicas y productos para el integrar la AI, considerando un enfoque ágil y centrado en el usuario.

Técnicas	Metodologías para la AI	A	D	E	I	G
Diagrama de Afinidad	(Lamar, 2001; Reichenauer, 2005)		X			
Investigación de Antecedentes	(Rosenfeld et al., 2015)	X				
Benchmarking	(Rosenfeld et al., 2015; Reichenauer, 2005)	X	X			X
Clasificación de Tarjetas	(Rosenfeld et al., 2015; Lamar, 2001; Reichenauer, 2005; Toub, 2000)	X	X	X		X
Inspección de Consistencia	(Reichenauer, 2005)		X			
Evaluación Consolidada	(Reichenauer, 2005)	X	X			
Diagramación	(Lamar, 2001)		X			
Modelo Entidad-Relación	(Reichenauer, 2005)			X		
Análisis de Retroalimentación	(Rosenfeld et al., 2015, 2006; Toub, 2000; Reichenauer, 2005)			X		X
Estudio de Campo	(Reichenauer, 2005)	X				X
Grupos de Discusión	(Rosenfeld et al., 2015; Reichenauer, 2005)	X				X
Puntuación de Bondad	(Toub, 2000)			X		

Evaluación Heurística	(Rosenfeld et al., 2015; Toub, 2000; Reichenauer, 2005)	X		X		
Patrones de Diseño de Interfaz	(Reichenauer, 2005)	X	X		X	
Entrevistas	(Rosenfeld et al., 2015; Sharlin et al., 2009; Reichenauer, 2005)	X		X		
Prototipado de Baja Fidelidad	(Reichenauer, 2005)	X	X			
Reuniones	(Rosenfeld et al., 2015; Reichenauer, 2005)	X			X	
Prototipo de Maqueta	(Reichenauer, 2005)	X	X			
Diseño Participativo	(Rosenfeld et al., 2015; Reichenauer, 2005)		X		X	
Personas	(Rosenfeld et al., 2015; Lamar, 2001; Reichenauer, 2005)	X	X			
Evaluación de Previsibilidad y Eficiencia	(Toub, 2000)			X		
Cuestionarios	(Lamar, 2001; Toub, 2000)	X		X		
Escenarios	(Rosenfeld et al., 2015; Reichenauer, 2005)	X	X			
Verificación de la Sentencia Acelerada	(Toub, 2000)			X		
Evaluación de la Estructura dirigida por el Patrocinador	(Toub, 2000)			X		
Storyboards	(Reichenauer, 2005)			X		
Evaluación de la Estructura	(Reichenauer, 2005)		X			
Evaluación de la Usabilidad	(Reichenauer, 2005)			X		
Encuesta	(Toub, 2000)	X		X		
Talleres	(Rosenfeld et al., 2015; Reichenauer, 2005)	X			X	

**Tabla 15:** Técnicas recomendadas para integrar la AI centrada en los usuarios y ágil.

## 2.3. Metodologías Ágiles

Una vez descritas y analizadas las diferentes propuestas para la Arquitectura de la Información (sección 2.2), en esta sección se describen, de manera sistemática, las metodologías ágiles más importantes, con el fin de revisar su viabilidad para integrar el Diseño Centrado en el Usuario. El objetivo de esta sección es identificar un marco de trabajo ágil, adecuado para el desarrollo centrado en el usuario de la Arquitectura de la Información, y corroborar además la hipótesis de partida **H1.3**: La mayoría de las metodologías ágiles presentan dificultades para integrar el Diseño Centrado en el Usuario y sólo consideran a los usuarios finales de manera parcial durante el desarrollo de software.

Asimismo, con la realización de este estudio también se busca responder las siguientes preguntas de investigación (definidas en la sección 1.2):

**PI<sub>7</sub>**: ¿Qué metodologías de desarrollo de software son comúnmente consideradas como ágiles?

**PI<sub>8</sub>**: ¿Qué enfoque de desarrollo y modelo de proceso son los más habituales en las metodologías ágiles?

**PI<sub>9</sub>**: ¿Qué metodología ágil permite una mejor integración del Diseño Centrado en el Usuario?

### 2.3.1. Definición

Las metodologías tradicionales de desarrollo de software se basan en procesos muy controlados por políticas y normas, y requieren de un seguimiento estricto del plan inicial de desarrollo, dificultando responder con flexibilidad ante los cambios del proyecto. Ante este escenario, las metodologías ágiles surgen como alternativa, para responder a la necesidad de desarrollar software con rapidez, en un entorno donde los requisitos cambian rápidamente. Estas metodologías permiten la entrega rápida de software de alta calidad, en alineación con las necesidades del cliente (Collier, 2011).

En estas metodologías, los requisitos se desarrollan en pequeños incrementos con una planificación mínima a largo plazo. De este modo, en cada iteración, el equipo de trabajo realiza un ciclo completo de desarrollo de software, incluyendo planificación, análisis de requisitos, diseño, codificación, pruebas unitarias y pruebas de aceptación. Esto permite minimizar el riesgo global y adaptar el proyecto a los cambios más rápidamente (Martin, 2003; Grey, 2011).

### 2.3.2. El Manifiesto Ágil

El manifiesto ágil es una compilación de los valores y principios, en los que se sustentan las metodologías ágiles. En este manifiesto, se describen cuatro valores básicos y 12 principios que detallan las mejores prácticas para el desarrollo de software (Beck et al., 2001). A continuación, se presentan los valores y principios del manifiesto ágil:

**a) Valores**

- *A los individuos y su interacción, por encima de los procesos y las herramientas.* Este valor pone de relieve la importancia de crear equipos de trabajo motivados, por encima de centrarse en definir el proceso y las herramientas, y esperar a que el equipo se adapte a ellas.
- *El software que funciona, por encima de la documentación exhaustiva.* La prioridad está en crear incrementos de valor del proyecto en cada iteración y generar sólo la documentación esencial del proyecto.
- *La colaboración con el cliente, por encima de la negociación contractual.* Se plantea que exista una colaboración cercana entre el equipo de trabajo y los clientes, por encima de priorizar el seguimiento de contratos o planes para el desarrollo de software.
- *La respuesta al cambio, por encima del seguimiento de un plan.* La planificación del proyecto debe ser flexible y responder rápidamente a los cambios que se producen.

En base a estos valores, los firmantes del manifiesto ágil (Beck et al., 2001) redactaron los siguientes principios:

**b) Principios**

1. *La prioridad está en satisfacer al cliente mediante la entrega temprana y continua de software con valor.*
2. *Se acepta que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.*
3. *Se entrega software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.*
4. *Los responsables de negocio y los desarrolladores trabajan juntos, de forma cotidiana, durante todo el proyecto.*
5. *Los proyectos se desarrollan en torno a individuos motivados. Se deben proveer el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.*
6. *El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.*
7. *El software funcionando es la medida principal de progreso.*
8. *Los procesos ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios deben ser capaces de mantener un ritmo constante de forma indefinida.*
9. *La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.*



10. *La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.*
11. *Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.*
12. *A intervalos regulares, el equipo debe reflexionar sobre cómo ser más efectivo, para a continuación ajustar y perfeccionar su comportamiento en consecuencia.*

Estos principios proveen las normas generales que deben seguir las metodologías ágiles, planteando confiar en las interacciones entre los individuos, facilitar el cambio en el proceso de desarrollo de acuerdo a las exigencias de los entornos, minimizar la documentación, utilizar el software como medida de la rapidez de la generación de los resultados, proporcionar una retroalimentación rápida, entre otros.

### **2.3.3. Descripción de las Metodologías Ágiles**

Una vez presentados los lineamientos generales que deben seguir las metodologías ágiles, a continuación, se describen las principales características de las metodologías ágiles más comunes, con el objetivo de posteriormente analizar e identificar en qué medida las metodologías ágiles permiten integrar el Diseño Centrado en el Usuario.

Las metodologías ágiles que se describirán corresponden a: Scrum (Schwaber y Beedle, 2002), Programación Extrema (XP) (Beck, 2000), Desarrollo de Software Adaptativo (ASD) (Highsmith, 2000), Desarrollo de Software Dirigido por Funcionalidades (FDD) (Coad et al., 1999), Método de Desarrollo de Sistemas Dinámicos (DSDM) (Stapleton, 1997), y el Método Kanban (Anderson, 2003). Estas metodologías se han seleccionado porque corresponden a las indicadas de forma más recurrente por los documentos más representativos de la literatura ágil (Beck et al., 2001; Cockburn y Highsmith, 2001; Highsmith y Cockburn, 2001; Boehm, 2002).

Por otro lado, se revisarán las metodologías anteriores de manera sistemática, utilizando un conjunto de aspectos que permiten identificar los componentes y las características más representativas de cada metodología. Estos aspectos corresponden a: *Proceso*, *Roles y Responsabilidades*, *Prácticas*, y *Valores y Principios*. El aspecto *Proceso* se refiere a la descripción de las fases en el ciclo de vida a través del cual se produce el software. El aspecto *Roles y Responsabilidades* se refiere a la asignación de las funciones a través de las cuales la producción de software se lleva a cabo en un equipo de desarrollo. El aspecto *Prácticas* se refiere a las actividades y los productos de trabajo que se definen para utilizarlos en el proceso. Finalmente, el aspecto *Valores y Principios* se refieren a los lineamientos más importantes de las metodologías.

A continuación, se presenta la descripción de las metodologías ágiles, de acuerdo a los diferentes aspectos definidos, con el fin de posteriormente analizarlas comparativamente:

### 2.3.3.1. Scrum

Scrum es un proceso ágil de desarrollo software, creado por Schwaber y Sutherland (Sutherland y Schwaber, 1995), basado principalmente en la gestión y el control de los procesos. El marco de trabajo Scrum consiste en equipos de trabajo, roles, eventos, artefactos y reglas asociadas (Schwaber y Beedle, 2002). Las reglas de Scrum relacionan los eventos, roles y artefactos, gobernando las relaciones e interacciones entre ellos. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo (Verheyen, 2013).

#### a) Proceso

El proceso de Scrum se compone de cuatro fases: *Gestión de Requisitos*, *Planificación del Sprint*, *Ejecución del Sprint* y *Clausura*. Las tres primeras fases se repiten hasta completar los suficientes requisitos del proyecto, agotando el presupuesto o llegando a una fecha límite. A continuación, se describen las fases de Scrum:

**Gestión de Requisitos:** En esta fase, se definen los requisitos del proyecto que pueden proceder de los clientes, de las divisiones de ventas, de atención al cliente o de los desarrolladores de software. No obstante, el *Dueño del Producto* (ver el apartado de *Roles y Responsabilidades* de esta sección) es el responsable final de la correcta gestión de los requisitos. Para apoyar esta tarea, el *Dueño del Producto* utiliza la *Lista del Producto*, que es una lista ordenada de todo lo que se podría necesitar en el proyecto y es la única fuente de requisitos para cualquier cambio a realizarse en el producto. El *Dueño del Producto* es el responsable de la Lista del Producto, incluyendo su contenido, disponibilidad y ordenación.

**Planificación del Sprint:** En esta fase, se identifican los requisitos que el *Equipo de Desarrollo* (ver el apartado de *Roles y Responsabilidades* de esta sección) realizará en cada *Sprint* (ver el apartado de *Prácticas* de esta sección). El *Equipo de Desarrollo* divide los requisitos seleccionados en tareas específicas de desarrollo que posteriormente se desarrollan durante la siguiente fase (*Ejecución del Sprint*).

**Ejecución del Sprint:** En esta fase, se desarrolla el conjunto de requisitos comprometidos por el *Equipo de Desarrollo*, con el fin de generar un incremento del proyecto (esto corresponde a la suma de todos los requisitos de la *Lista del Producto*, completados durante un *Sprint*, y el valor de los incrementos de todos los *Sprints* anteriores). Al final de esta fase, el nuevo incremento debe estar en condiciones para que se pueda utilizar y cumplir con la definición de “*Terminado*” del equipo Scrum. La definición de “*Terminado*” es un acuerdo del equipo Scrum para determinar el momento en el que un requisito puede presentarse al Dueño del Producto o a los Stakeholders.

**Clausura:** En esta fase, se realizan las tareas para cerrar el proyecto. Se entra en esta fase cuando no hay más requisitos ni temas que se deban implementar. En esta fase, el sistema se encuentra listo para la liberación, y la preparación para esto incluye tareas relacionadas con la integración, las pruebas del sistema y la documentación.

### **b) Roles y Responsabilidades**

Los equipos Scrum son auto-organizados y multifuncionales, y están compuestos por el Dueño del Producto (*Product Owner*), el Equipo de Desarrollo (*Development Team*), un Scrum Master y diferentes Stakeholders. A continuación, se describe el equipo Scrum:

**El Dueño del Producto:** Es el responsable de maximizar el valor del producto y del trabajo del Equipo de Desarrollo. El Dueño del Producto es la persona responsable de gestionar la Lista del Producto (*Product Backlog*) mediante las siguientes tareas (Schwaber y Sutherland, 2011):

- Expresar claramente los elementos de la Lista del Producto;
- Ordenar los elementos en la Lista del Producto para alcanzar los objetivos y misiones de la mejor manera posible;
- Asegurar que el Equipo de Desarrollo entiende los elementos de la Lista del Producto a nivel necesario.

**El Equipo de Desarrollo:** Son los profesionales que tienen la responsabilidad de entregar el incremento de producto “Terminado”, que potencialmente se pueda poner en producción al final de cada *Sprint*. El Equipo de Desarrollo tiene las siguientes características:

- Son auto-organizados. El Equipo de Desarrollo es el encargado de convertir los elementos de la Lista del Producto en incrementos de funcionalidad potencialmente desplegables;
- Los Equipos de Desarrollo son multifuncionales, contando con todas las habilidades necesarias para crear un incremento de producto;
- Los miembros individuales del Equipo de Desarrollo pueden tener habilidades especializadas y áreas en las que estén más especializados, pero la responsabilidad recae en el Equipo de Desarrollo como un todo.

**El Scrum Master:** Es el responsable de asegurar que Scrum sea entendido y adoptado. El trabajo principal del Scrum Master es eliminar los impedimentos que imposibilitan que el equipo Scrum alcance el objetivo del *Sprint*.

**Stakeholders:** Son las personas (clientes, trabajadores, proveedores, vendedores, etc.) que hacen posible el proyecto. Los Stakeholders no se involucran frecuentemente en el proceso Scrum, no obstante, se deben tener en cuenta porque son afectados por las decisiones del proyecto.

### **c) Prácticas**

Scrum proporciona un conjunto de prácticas y herramientas de gestión en sus diferentes fases para evitar el caos producido por la imprevisibilidad y la complejidad. Algunas de las prácticas y eventos de Scrum corresponden a (Schwaber y Sutherland, 2011):

**El Sprint:** Es un bloque de tiempo de un mes o menos, durante el cual se crea un incremento de producto “Terminado”, utilizable y potencialmente desplegable.

**Reunión de Planificación del Sprint** (Sprint Planning Meeting): El trabajo a realizar durante el Sprint se planifica en la Reunión de Planificación del Sprint. Durante ésta, el Equipo de Desarrollo identifica los requisitos y las tareas de desarrollo para el siguiente Sprint, basándose en las prioridades y preferencias establecidas por el Dueño del Producto.

**Scrum Diario** (Daily Scrum): Es una reunión para discutir los avances, las metas y los impedimentos que enfrenta cada uno de los miembros del Equipo de Desarrollo durante la Ejecución del Sprint. Durante la reunión, cada miembro del Equipo de Desarrollo explica lo siguiente:

¿Qué hice ayer que ayudó al Equipo de Desarrollo a lograr el objetivo del Sprint?

¿Qué haré hoy para ayudar al Equipo de Desarrollo a lograr el objetivo del Sprint?

¿Veo algún impedimento que evite que el Equipo de Desarrollo o yo logremos el objetivo del Sprint?

**Revisión del Sprint** (Sprint Review): Al final del Sprint, se lleva a cabo su revisión para inspeccionar el incremento y adaptar la Lista del Producto si fuese necesario.

**Retrospectiva del Sprint** (Sprint Retrospective): La Retrospectiva del Sprint es una oportunidad para el equipo Scrum de inspeccionarse a sí mismo y crear un plan de mejoras, para que se aborden durante el siguiente Sprint.

#### **d) Valores y Principios**

El trabajo realizado en Scrum requiere de un conjunto de valores, como base para el proceso y las interacciones del equipo. En Scrum, se proponen los siguientes valores (Schwaber y Beedle, 2002):

**Enfoque:** Este valor se refiere a que el equipo Scrum debe concentrarse para lograr entregar un incremento al final de cada Sprint. Este valor también significa trabajar en un proyecto a la vez.

**Respeto:** Los equipos de Scrum de alto desempeño se basan en el respeto para alcanzar el éxito en los proyectos. El respeto a los compañeros de equipo puede hacer o deshacer un proyecto.

**Compromiso:** El equipo Scrum hace un compromiso con la organización y el uno al otro, durante cada Reunión de Planificación del Sprint. Al final de la Planificación del Sprint, cada miembro del equipo debe tener el mismo nivel de comprensión acerca de lo que el equipo se compromete a llevar a cabo durante ese Sprint.

**Coraje:** El coraje es la capacidad para enfrentar dificultades a pesar de los temores. El alivio es que estos temores es una de las mejores maneras para que el equipo Scrum y la organización puedan ayudar a los miembros a que sean valientes.

**Apertura:** La apertura permite ser más receptivo a nuevas ideas. El equipo Scrum debe tener la voluntad para recibir nuevas ideas y formas de pensar, con el objetivo de lograr un equipo de alto rendimiento.

### 2.3.3.2. Programación extrema (XP)

A diferencia de Scrum, que está enfocada en la gestión de los procesos de desarrollo de software, XP es una metodología centrada en el diseño del producto y en potenciar las relaciones interpersonales como clave para el éxito del desarrollo de software. Esta metodología fue creada por Beck (Beck, 2000) y sus principales objetivos son promover el trabajo en equipo, preocuparse por el aprendizaje de los desarrolladores y propiciar un buen clima de trabajo.

#### a) Proceso

En XP los requisitos se implementan a través de un modelo iterativo e incremental, donde tanto el cliente como el programador aprenden. A continuación, se describen las diferentes fases del ciclo de vida de XP:

**Fase de Exploración:** En esta fase, los clientes señalan, a grandes rasgos, los requisitos que son de interés para la primera entrega del producto. Además, los diferentes miembros del equipo XP examinan la tecnología y construyen un prototipo inicial, con el fin de revisar las alternativas tecnológicas.

**Fase de Planificación:** En esta fase se priorizan los requisitos y se acuerda el alcance del entregable. Los programadores estiman el esfuerzo necesario para implementar cada requisito, y en base a esto definen el cronograma del proyecto.

**Iteraciones para la Primera Liberación:** En esta fase, se diseña la arquitectura general del sistema, en base a los requisitos que facilitan esta construcción. Posteriormente, el cliente selecciona los requisitos que se implementarán en cada iteración, de acuerdo a sus preferencias y limitaciones de tiempo. Por último, al final de cada iteración, se realizan diferentes pruebas funcionales que son dirigidas por el cliente.

**Fase de Producción:** En esta fase, se realizan evaluaciones y pruebas de las funcionalidades del sistema, con el fin de entregar una versión utilizable para el cliente. Además, los resultados de estas evaluaciones se documentan, para proveer sugerencias para la *Fase de Mantenimiento*.

**Fase de Mantenimiento:** En esta fase, se realizan diferentes tareas para conservar las funcionalidades del sistema y satisfacer las tareas del cliente.

**Fase de Muerte:** En esta fase, se entra cuando el cliente no tiene más requisitos para incluir en el sistema.

#### b) Roles y Responsabilidades

En XP se definen diferentes roles, con el fin de cumplir las diferentes tareas y propósitos de las fases del ciclo de vida de XP. A continuación, se presentan los roles propuestos en XP (Rosenberg y Stephens, 2008):

**Programador:** Es la persona que define las pruebas unitarias y genera el código del sistema.

**Cliente:** Es la persona que especifica los requisitos y las pruebas funcionales para validar la implementación del sistema. Además, determina la prioridad de los requisitos y resuelve cuáles se realizan en cada iteración.

**Probador:** Es la persona que apoya al cliente para definir las pruebas funcionales. Además, ejecuta las pruebas, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

**Rastreador:** Es la persona encargada de realizar el seguimiento del sistema, con el fin de proveer retroalimentación al equipo.

**Entrenador:** Es la persona responsable del proceso completo, y de guiar a los miembros del equipo para alcanzar el proceso correctamente.

**Consultor:** Es una persona externa del equipo que posee conocimientos específicos en algún tema necesario para el proyecto. El objetivo de este rol es auxiliar al equipo, para solucionar un problema específico.

**Gestor:** Es la persona que coordina las labores de los programadores y los clientes.

### c) Prácticas

XP provee un conjunto de prácticas, enfocadas principalmente en buscar la eficiencia en la generación de código fuente. A continuación, se describen las prácticas de la metodología XP (Ashmore y Runyan, 2014):

**La Planificación del Juego:** El alcance de la siguiente versión se define por las consideraciones del negocio y las estimaciones técnicas. Por lo tanto, se debe determinar rápidamente el alcance de la próxima versión, combinando las prioridades del negocio y las estimaciones técnicas.

**Versiones Pequeñas:** Esta práctica consiste en liberar nuevas versiones del sistema en ciclos cortos de desarrollo, con el fin de fomentar la participación de los clientes.

**Metáfora:** Esta práctica consiste en definir una metáfora para el sistema, con el fin de guiar todo el desarrollo con una historia compartida de cómo funciona el sistema.

**Pruebas Continuas:** Esta práctica consiste en validar el desarrollo del sistema, mediante la generación continua de pruebas unitarias.

**Refactorización:** Esta práctica consiste en reestructurar el sistema, alterando sólo su estructura interna, sin cambiar su comportamiento externo.

**Programación por Parejas:** Esta práctica consiste en generar la producción del código con dos programadores en una sola computadora.

**Integración Continua:** Esta práctica consiste en integrar y construir el sistema muchas veces al día, cada vez que se complete una tarea.

**Historias de Usuario:** Es una representación de alto nivel de los requisitos de un proyecto software desde el punto de vista del usuario o del cliente.

**Cliente en el sitio:** Esta práctica consiste en incluir a un usuario real en el equipo, disponible a tiempo completo para responder a las preguntas del equipo.

**Estándares de Codificación:** Esta práctica consiste en escribir el código fuente, de acuerdo con las normas que enfatizan la comunicación a través del código.

#### **d) Valores y Principios**

En XP se proponen cinco valores, con el objetivo de guiar el desarrollo y promover el equilibrio y el apoyo mutuo (Beck, 2000). A continuación, se presentan los valores de la metodología XP:

**Comunicación:** Este valor se refiere a la necesidad de promover la comunicación directa y continua entre clientes y desarrolladores, con el objetivo de crear un sentido de equipo y de cooperación efectiva.

**Simplicidad:** Este valor se refiere a desarrollar sólo el sistema que realmente se necesita, con el fin de promover el desarrollo simple, y eliminar los requisitos innecesarios.

**Retroalimentación:** Este valor se relaciona con los dos valores anteriores (comunicación y simplicidad). Es decir, una mayor retroalimentación genera una mejor comunicación, y cuanto más simple es un sistema, más fácil es de probar.

**Valentía:** La valentía se refiere al hecho de descartar las soluciones que fallan y buscar otras nuevas con respuestas reales.

#### **2.3.3.3. Desarrollo de Software Adaptativo (ASD)**

ASD es un enfoque ágil y adaptativo para el desarrollo de proyectos software de alta velocidad y alto cambio. Esta metodología fue creada por Highsmith (Highsmith, 2000), y se originó en el contexto de los sistemas complejos y la adaptación continua del proceso de trabajo. ASD proporciona un proceso más ligero que Scrum y XP, y se centra en la colaboración humana y la organización del equipo para desarrollar software y sistemas muy complejos. Asimismo, en ASD se pone más énfasis en los resultados y su calidad que en las tareas o el proceso, utilizados para producir los resultados.

##### **a) Proceso**

En ASD no hay un ciclo de vida estático, sino que iterativo e incremental, donde cada uno de esos ciclos se puede modificar al tiempo que otro es ejecutado (Highsmith y Cockburn, 2001). El proceso de ASD está enfocado en el aprendizaje y colaboración entre los desarrolladores y los clientes. Este proceso está compuesto por tres fases: *Especulación, Colaboración y Aprendizaje*.

**Especulación:** Esta fase consiste en definir los lineamientos generales del proyecto y realizar su planificación adaptativa. Ésta última consiste en tratar las desviaciones de la planificación como guías hacia la solución correcta, y no como errores que se deben corregir. A continuación, se presentan los pasos propuestos para realizar esta fase (Highsmith, 2000):

1. Inicio para determinar la misión del proyecto.
2. Fijación del marco temporal del proyecto.
3. Definición del número de iteraciones y la duración de cada una.

4. Determinación del objetivo de cada iteración.
5. Asignación de las funcionalidades de cada iteración.

**Colaboración:** Esta fase consiste en desarrollar las funcionalidades del sistema y gestionar el producto. Mientras que el equipo técnico trabaja en entregar funcionalidades del sistema, los encargados del proyecto facilitan la colaboración y las actividades simultáneas de desarrollo.

**Aprendizaje:** Esta fase consiste en examinar el producto desde la perspectiva del cliente y la calidad técnica, y también en inspeccionar el desempeño del equipo del proyecto. Hay cuatro categorías generales de cosas para aprender, al final de cada iteración de desarrollo:

- Calidad del producto desde el punto de vista del cliente.
- Calidad del producto desde el punto de vista de los desarrolladores.
- Funcionalidad desarrollada.
- Estado del proyecto.

#### **b) Roles y Responsabilidades**

ASD no describe estructuras de equipo en detalle. En esta metodología, cada proyecto debe definir las responsabilidades y las relaciones del equipo de trabajo, de acuerdo al contexto y las necesidades del proyecto.

#### **c) Prácticas**

ASD presenta un conjunto de prácticas para apoyar la planificación adaptativa y el aprendizaje. A continuación, se presentan las prácticas de ASD (Highsmith, 2013):

**Planificación del Ciclo:** En esta práctica, se definen tres pasos para planificar un proyecto de manera adaptativa. Primero, el equipo debe definir los componentes del producto. Posteriormente, estos componentes se deben asignar a un ciclo de desarrollo y, finalmente, se deben planificar las actividades necesarias para cada ciclo.

**Revisiones del Ciclo:** Esta práctica consiste en proporcionar, al final de cada ciclo de desarrollo, un ciclo de reflexión para determinar el estado del proyecto.

**Grupos de Debate de Clientes:** Es una herramienta de investigación que tiene una doble finalidad: permitir a los miembros del equipo de desarrollo recopilar información de los clientes sobre el producto intermedio, y ayudar a los desarrolladores a construir mejores relaciones con los clientes.

**Inspecciones de Software:** Es una revisión por pares de cualquier producto de trabajo, realizada por personas capacitadas, con el fin de buscar defectos mediante un proceso bien definido.

**Postmortem:** Esta práctica consiste en evaluar los éxitos y los desafíos del proyecto como una oportunidad para reorientar el plan.



#### d) Valores y Principios

En ASD, se definen tres valores para promover la gestión adaptativa. Estos valores tratan de crear un ambiente para responder al desafío de los proyectos extremos, especialmente el desafío de cambio. A continuación, se presentan los tres valores de ASD:

**Liderazgo:** Los líderes deben crear entornos en los que las ideas florezcan, se tomen riesgos, y los errores sean vistos como oportunidades de aprendizaje.

**Colaboración:** La colaboración, como una actividad base en el ciclo de vida de desarrollo adaptativo, se centra en las relaciones interpersonales, dentro de un equipo de proyecto.

**Rendición de Cuentas:** Este valor consiste en garantizar que los resultados correctos se entreguen a las personas adecuadas, en el momento oportuno.

#### 2.3.3.4. Desarrollo Dirigido por Funcionalidades (FDD)

FDD es un enfoque para el desarrollo de software que combina algunas de las ventajas de las metodologías Scrum y XP, junto con un conjunto de las mejores prácticas reconocidas por la industria, especialmente las que son dirigidas desde la perspectiva de la funcionalidad. FDD provee un proceso ligero, parecido a XP, y su principal objetivo es entregar un software tangible, con énfasis en las fases de diseño y construcción (Hunt, 2006).

##### a) Proceso

FDD proporciona un proceso iterativo con iteraciones cortas, menor a tres semanas, que producen un software funcional. A continuación, se describen las cinco fases de FDD:

**Desarrollo de un Modelo General:** Esta fase consiste en construir un modelo general del proyecto, considerando la visión, el contexto y los requisitos del sistema. Este modelo general se divide en áreas que se analizan detalladamente, y se elabora un diagrama de las clases por cada área.

**Construcción de una Lista de Funcionalidades:** Esta fase consiste en desarrollar una lista con las funcionalidades del sistema, basado en el conocimiento reunido en la fase inicial (Desarrollo de un Modelo General). Las funcionalidades de la lista se dividen en funcionalidades más pequeñas, para un mejor entendimiento del sistema.

**Planificación por Funcionalidades:** Esta fase consiste en crear un plan de desarrollo que es realizado por el *Propietario de Clases*. Para llevar a cabo esta tarea, se ordenan las funcionalidades de acuerdo a su prioridad y dependencia, y se asignan a los *Programadores Jefes*.

**Diseño por Funcionalidades:** Esta fase consiste en diseñar las funcionalidades asignadas a los *Programadores Jefes*. El diseño de estas funcionalidades se realiza mediante un proceso iterativo, escogiendo las funcionalidades en cada iteración. Esta fase se compone de un conjunto de actividades, relacionadas con la inspección del diseño, la codificación, las pruebas unitarias, la integración y la inspección de código.

**Construcción por Funcionalidades:** Esta fase consiste en construir todas funcionalidades, diseñadas en la fase anterior.

#### **b) Roles y Responsabilidades**

Los roles en FDD hacen énfasis en resolver los aspectos técnicos del proyecto. A continuación, se describen los diferentes roles de FDD (Palmer y Felsing, 2002):

**Gerente de Proyecto:** Es el responsable administrativo y financiero del proyecto. Algunas de sus tareas incluyen proteger el equipo del proyecto de distracciones externas y permitir que los miembros del equipo trabajen juntos, dotándolos de unas condiciones de trabajo adecuadas.

**Arquitecto Jefe:** Es el responsable del diseño global del sistema y de ejecutar las sesiones de diseño de los talleres celebrados con el equipo.

**Director de Desarrollo:** El director de desarrollo conduce las actividades de desarrollo diarias y resuelve los conflictos que puedan ocurrir dentro del equipo.

**Programador Jefe:** El jefe de programación es un desarrollador con experiencia que participa en el análisis de los requisitos y en el diseño de los proyectos.

**Propietario de Clases:** El propietario trabaja bajo la dirección del programador jefe en las tareas de diseño, codificación, pruebas y documentación. Este rol es el responsable del desarrollo de las clases asignadas.

**Administrador de Herramientas:** Es un rol para la construcción de pequeñas herramientas para los equipos de desarrollo, de prueba y de conversión de datos en el proyecto.

**Administrador del Sistema:** La tarea de un administrador de sistemas es configurar, administrar y solucionar problemas de los servidores, la red de estaciones de trabajo, los entornos de desarrollo y las pruebas, utilizadas por el equipo del proyecto.

#### **c) Prácticas**

FDD presenta una serie de prácticas, con el fin de apoyar el diseño y la construcción de los requisitos. A continuación, se describen las mejores prácticas que componen FDD (Chowdhury y Huda, 2011):

**Modelado de Objetos de Dominio:** Esta práctica consiste en explorar y explicar el dominio del problema a resolver. El modelo de objetos de dominio resultante proporciona un marco general para añadir funcionalidades.

**Desarrollo por Funcionalidades:** Esta práctica consiste en descomponer las funcionalidades complejas en funcionalidades más pequeñas, hasta que cada problema sea lo suficientemente pequeño para que se pueda tratar como una funcionalidad.

**Propiedad Individual de la Clase:** Esta práctica consiste en asignar piezas distintas o grupos de clases a un solo propietario. El propietario es responsable de la coherencia, el rendimiento y la integridad conceptual de la clase.

**Inspecciones:** Esta práctica consiste en realizar inspecciones para garantizar la calidad del diseño y del código.

**Gestión de la Configuración:** Esta práctica consiste en mantener un historial de los cambios en las clases y de los códigos fuente de las funcionalidades, completadas hasta la fecha.

**a) Valores y Principios**

FDD no define valores ni principios.

**2.3.3.5. Método de Desarrollo de Sistemas Dinámicos (DSDM)**

DSDM provee un marco de trabajo para el desarrollo ágil de software, basado en las mejores prácticas y lecciones aprendidas en la industria, con una participación activa de los usuarios (Stapleton, 1997). Esta metodología se publicó en 1995 por el Consorcio DSDM (Stapleton, 1999) y se centra en ofrecer una eficiente gestión de los proyectos, similar a Scrum, y un control fuerte en el ciclo de vida del proyecto.

**a) Proceso**

DSDM utiliza un ciclo de vida iterativo e incremental, compuesto por cinco fases: *el Estudio de Viabilidad, el Estudio de Negocios, Iteración del Modelado Funcional, Iteración de Diseño y Desarrollo e Implementación*. Las dos primeras fases se realizan de manera secuencial y se hacen sólo una vez, mientras que las últimas tres fases son iterativas e incrementales. A continuación, se describen las fases de DSDM:

**El Estudio de Viabilidad:** Esta fase consiste en estudiar la viabilidad del proyecto para que se pueda implementar mediante DSDM. En este estudio, se realizan diferentes evaluaciones, relacionadas con los costes y la viabilidad técnica del proyecto.

**El Estudio de Negocios:** En esta fase, se extiende el estudio de viabilidad. Después de que el proyecto se ha considerado factible para el uso de DSDM, se estudian y analizan los requisitos del negocio y las restricciones técnicas.

**Iteración del Modelo Funcional:** Los requisitos, identificados en las etapas anteriores, se convierten en un modelo funcional (prototipo). Esta fase está compuesta por cuatro actividades:

- *Identificación del Prototipo Funcional:* En esta actividad se determinan las funcionalidades a implementar en el prototipo de la iteración.
- *Acuerdo del Programa:* En esta actividad se debe acordar cómo y cuándo desarrollar las funcionalidades identificadas en la actividad anterior.
- *Creación de un Prototipo Funcional:* En esta actividad se desarrolla el prototipo, de acuerdo a las funcionalidades identificadas.
- *Revisión del Prototipo:* En esta actividad se evalúa el prototipo desarrollado.

**Iteración de Diseño y Desarrollo:** El objetivo de esta fase es integrar los componentes funcionales de la fase anterior en un sistema que satisfaga las necesidades de los usuarios. Esta fase está compuesta por las cuatro actividades de la fase anterior (Iteración del Modelo Funcional).

**Implementación:** Esta fase consiste en evaluar el sistema, entregar la documentación a los usuarios y capacitar a los futuros usuarios. Esta fase contiene cuatro actividades:

- *Conformidad de los Usuarios y las Directrices:* En esta actividad, los usuarios finales aprueban el sistema testado para la implementación y se generan las directrices del proyecto.
- *Entrenamiento de los Usuarios:* En esta actividad, se entrena a los usuarios finales en el uso del sistema.
- *Implementación:* En esta actividad, se implementa el sistema, probado en la ubicación de los usuarios finales.
- *Revisión del Negocio:* En esta actividad, se revisa el efecto del sistema, implementado en la empresa, con el fin de verificar si el sistema cumple con los objetivos, establecidos al comienzo del proyecto.

#### **b) Roles y Responsabilidades**

DSDM define diferentes roles, relacionados con los usuarios y desarrolladores. A continuación, se describen los roles más importantes de DSDM (Stapleton, 1997):

**Los Desarrolladores y los Desarrolladores de Alto Nivel:** Los roles del desarrollador y del desarrollador de alto nivel cubren todo el personal de desarrollo, ya sea de los analistas, diseñadores, programadores o probadores.

**Coordinador Técnico:** Define la arquitectura del sistema y es responsable de la calidad técnica en el proyecto.

**Usuario Embajador:** Es el encargado de transmitir el conocimiento de los usuarios dentro del proyecto y de difundir la información sobre el avance del proyecto a otros usuarios.

**Usuario Asesor:** Es el usuario que representa un punto de vista importante para el proyecto.

**Visionario:** Es el usuario que tiene la percepción más precisa de los objetivos de negocio del sistema y del proyecto.

**Patrocinador Ejecutivo:** Es la persona de la organización que tiene la autoridad y responsabilidad financiera. Por tanto, el patrocinador ejecutivo tiene el máximo poder en la toma de decisiones.

#### **c) Prácticas**

Nueve prácticas definen la ideología y la base de toda la actividad en DSDM. A continuación, se describen las prácticas más importantes de esta metodología (DSDM, 2008):

**Participación Activa de los Usuarios Expertos:** Los usuarios expertos tienen que estar presentes durante todo el desarrollo del sistema para asegurar una retroalimentación oportuna y precisa.

**Equipos Facultados para Tomar Decisiones:** Esta práctica consiste en permitir a los equipos DSDM tomar decisiones, con el fin de promover ciclos de desarrollo rápidos.

**Entrega Frecuente de Productos:** Esta práctica consiste en promover la entrega frecuente de productos, con el fin de acortar los ciclos de entrega y obtener una retroalimentación de los usuarios más frecuente.

**Propósito del Negocio:** Esta práctica consiste en enfocar el proceso de desarrollo en satisfacer las necesidades principales de negocio. Es decir, el propósito del negocio es el criterio esencial para la aceptación de los entregables.

**Desarrollo Iterativo e Incremental:** El desarrollo iterativo e incremental es necesario para converger en una solución de negocio precisa. Los requisitos del sistema rara vez permanecen intactos desde el inicio de un proyecto hasta su final. Por lo tanto, al permitir que los sistemas evolucionen a través del desarrollo iterativo, los errores se pueden encontrar y corregir a tiempo.

**Cambios del Desarrollo son Reversibles:** Esta práctica consiste en que los cambios durante el desarrollo del producto deben ser reversibles, mediante el uso de iteraciones cortas.

#### **d) Valores y Principios**

DSDM presenta un conjunto de valores para fomentar la retroalimentación y considerar las necesidades del negocio. A continuación, se presentan los principios que sustentan la metodología DSDM (Cooke, 2012):

**Centrarse en las Necesidades del Negocio:** Este valor consiste en que los principales criterios para la aceptación de una entrega del sistema son las necesidades actuales del negocio.

**Colaboración:** La participación de los usuarios expertos es la clave principal en la gestión del proyecto. Los usuarios expertos y desarrolladores comparten un lugar de trabajo, para que las decisiones se puedan hacer en colaboración y de forma rápida.

**Desarrollo Iterativo:** Este valor consiste en adoptar un enfoque iterativo para la construcción de todos los productos.

**Comunicación Continua y Clara:** Con el fin de conseguir eficiencia y eficacia, se requiere la comunicación y la cooperación entre todos los interesados del proyecto.

#### **2.3.3.6. Método Kanban (MK)**

El origen de Kanban está en los procesos de producción, ideado por Toyota en 1950. En este contexto, Kanban consiste en un sistema de señalización basado en tarjetas, con el fin de coordinar la reposición de piezas en sus líneas de montaje (Rasmusson, 2010). No obstante, en el contexto del desarrollo de software, el método Kanban se origina en 2004 y es introducido por (Anderson, 2010).

El objetivo principal del método Kanban es gestionar, de manera general, el proceso de realización de las tareas de los equipos de desarrollo de software. Así pues, este método promueve la posibilidad de visualizar el flujo de trabajo (mediante el *tablero Kanban*),

limitar el trabajo en progreso en cada etapa del flujo de trabajo y medir el tiempo del ciclo en los equipos de desarrollo de software (Ahmad et al., 2013).

El tablero Kanban es una herramienta para visualizar el trabajo que realiza cada miembro del equipo, que permite comunicar las prioridades e indicar los cuellos de botella. La prioridad es desarrollar sólo aquellos elementos solicitados y generar un flujo constante de los elementos de trabajo, liberados a los clientes.

El método Kanban es el más ligero de las metodologías ágiles, entregando pocas restricciones y dejando casi todo abierto. Las únicas exigencias son visualizar el flujo de trabajo y limitar el trabajo en progreso.

#### **a) Proceso**

El método Kanban no prescribe el uso de iteraciones y sólo entrega algunos lineamientos generales para conducir eficientemente el proceso (Anderson, 2010). Algunos de estos lineamientos corresponden a: centrarse en la calidad, reducir el trabajo en progreso, equilibrar la demanda en contra del rendimiento y atacar las fuentes de variabilidad para mejorar la previsibilidad.

#### **b) Roles y Responsabilidades**

El método Kanban no prescribe roles ni responsabilidades. En cambio, este método se basa en respetar las estructuras actuales de los equipos de trabajo.

#### **c) Prácticas**

Las seis prácticas básicas del método Kanban corresponden a (Skarin, 2015):

**Visualización del Flujo de Trabajo:** Esta práctica consiste en visualizar el flujo de trabajo, con el objetivo de comprender el progreso del trabajo y proveer la capacidad para intervenir en el equipo.

**Limitación del Trabajo en Progreso:** Esta práctica consiste en limitar el trabajo en curso del equipo de desarrollo de software, con el fin de equilibrar la demanda y la capacidad.

**Gestión del Flujo:** Esta práctica consiste en inspeccionar y medir el flujo de trabajo, con el fin de evaluar los aspectos positivos y negativos del sistema.

**Implementación de Ciclos de Retroalimentación:** Esta práctica consiste en implementar ciclos de retroalimentación, con el fin de conocer si el trabajo se realiza de forma correcta durante el desarrollo del producto.

#### **d) Valores y Principios**

El método Kanban plantea los siguientes principios básicos (Kirovska y Koceski, 2015):

**Reducción del Desperdicio:** Este principio consiste en reducir todo aquello que es superficial o secundario en el proyecto, y enfocarse en realizar solamente en lo requerido y necesario, pero sin perder la calidad de los productos desarrollados.

**Mejora Continua:** Se trata de promover la mejora continua del desarrollo de productos, en base a la gestión eficiente de las tareas.

**Flexibilidad:** Este principio consiste en priorizar las tareas, en base a las necesidades del momento, con el fin de proporcionar a los equipos la capacidad para responder a las tareas imprevistas.

### 2.3.3.7. Resumen de las Metodologías Ágiles

Como síntesis de la descripción de las metodologías ágiles presentadas anteriormente, en la Tabla 16 se presentan los principales rasgos de las metodologías ágiles. A continuación, se describen los valores de cada columna de la Tabla 16:

- *Enfoque de Desarrollo:* Esta columna se utiliza para identificar la visión filosófica sobre la que se construyó la metodología ágil (por ejemplo: enfoque estructurado, enfoque orientado a objetos, enfoque de modelado de información, enfoque centrado en el cliente, entre otros).
- *Fases Propuestas:* Se utiliza para identificar fases propuestas de las metodologías ágiles.
- *Modelo de Proceso de Desarrollo:* Esta columna sirve para identificar las secuencias de las etapas por las que el sistema evoluciona en la metodología ágil (por ejemplo: lineal, espiral, incremental, iterativo, entre otros).
- *Práctica de Desarrollo de Sistemas:* Se aplica para identificar las prácticas de desarrollo más destacadas para llevar a cabo la actividad de desarrollo.
- *Puntos Claves:* Esta columna se usa para describir los rasgos destacados de las metodologías ágiles.

Met. Ágiles	Enfoque de Desarrollo	Fases Propuestas	Modelo de Proceso de Desarrollo	Práctica de Desarrollo de Sistemas	Puntos Claves
Scrum	Enfoque ágil y centrado en el cliente.	<ul style="list-style-type: none"> <li>-Gestión de Requisitos.</li> <li>-Planificación del Sprint.</li> <li>-Ejecución del Sprint.</li> <li>-Clausura.</li> </ul>	Modelo incremental e iterativo.	<ul style="list-style-type: none"> <li>-Lista del Producto.</li> <li>-Sprint.</li> <li>-Lista de Pendientes del Sprint.</li> </ul>	El Equipo de Desarrollo es independiente, multifuncional y auto-organizado.
XP	Enfoque ágil y centrado en el cliente.	<ul style="list-style-type: none"> <li>-Fase de Exploración.</li> <li>-Fase de Planificación.</li> <li>-Iteraciones para la Primera Liberación.</li> <li>-Fase de Producción.</li> <li>-Fase de Mantenimiento.</li> <li>-Fase de Muerte.</li> </ul>	Modelo incremental e iterativo.	<ul style="list-style-type: none"> <li>-Programación por Parejas.</li> <li>-Refactorización.</li> <li>-Integración Continua.</li> </ul>	Rediseño continuo del sistema para mejorar el rendimiento y la capacidad de respuesta al cambio.
ASD	Enfoque ágil y centrado en el cliente.	<ul style="list-style-type: none"> <li>-Especulación.</li> <li>-Colaboración.</li> <li>-Aprendizaje.</li> </ul>	Modelo incremental e iterativo.	<ul style="list-style-type: none"> <li>-Inspecciones de Software.</li> <li>-Revisiones del Ciclo.</li> </ul>	Las organizaciones son vistas como sistemas adaptativos.



FDD	Enfoque ágil y centrado en el cliente.	<ul style="list-style-type: none"> <li>-Desarrollo de un Modelo General.</li> <li>-Construcción de una Lista de Funcionalidades.</li> <li>-Planificación por Funcionalidades.</li> <li>-Diseño por Funcionalidades.</li> <li>-Construcción por Funcionalidades.</li> </ul>	Modelo incremental e iterativo.	<ul style="list-style-type: none"> <li>-Modelado de Objetos de Dominio.</li> <li>-Propiedad Individual de la Clase.</li> <li>-Desarrollo por Funcionalidades.</li> </ul>	Método simplista con iteraciones muy cortas (de horas a dos semanas), enfocado en el diseño y la implementación de sistemas, dirigido por funcionalidades.
DSDM	Enfoque ágil y centrado en el cliente.	<ul style="list-style-type: none"> <li>-Estudio de Viabilidad.</li> <li>-Estudio de Negocios.</li> <li>-Iteración del Modelo Funcional.</li> <li>-Iteración de Diseño y Desarrollo.</li> <li>-Implementación.</li> </ul>	Modelo incremental e iterativo.	<ul style="list-style-type: none"> <li>-Entrega Frecuente de Productos.</li> <li>-Propósito del Negocio.</li> </ul>	Se definen roles, relacionados con los usuarios, y se faculta a los equipos para la toma de decisiones.
Método Kanban	Enfoque ágil y centrado en el cliente.	No prescribe el uso de iteraciones.	Modelo incremental e iterativo.	<ul style="list-style-type: none"> <li>-Visualizar el Flujo de Trabajo.</li> <li>-Limitar el Trabajo en Progreso.</li> </ul>	El desarrollo se basa en visualizar, limitar y medir el flujo de trabajo, con el fin de centrarse solamente en los elementos requeridos.

**Tabla 16:** Resumen de características de las metodologías ágiles.

Como se puede observar en la Tabla 16, todas las metodologías ágiles descritas tienen un enfoque de desarrollo ágil y centrado en el cliente. Aun cuando algunas de ellas proveen roles y prácticas para promover la participación de los usuarios, el objetivo principal está puesto en conducir el proceso a través de las preferencias de los clientes.

De igual manera, se puede observar que todas las metodologías ágiles tienen un modelo de proceso de desarrollo incremental e iterativo, para entregar incrementos de producto de software rápidamente y tan a menudo como sea posible, en un entorno en constante cambio.

En cuanto a los métodos y técnicas, se puede mencionar que estos están en concordancia con los valores y los principios descritos en cada una de las metodologías, siendo éstos últimos los que describen los lineamientos más importantes de cada una de ellas.

Otro aspecto a destacar, es la gran variedad de prácticas y técnicas, propuestas para llevar a cabo el desarrollo de software, desde el uso de prácticas reconocidas dentro del ámbito de desarrollo de software, hasta otras no tan populares. Estas prácticas están orientadas principalmente en proveer una mejor gestión de los cambios, generar una retroalimentación constante del cliente y facilitar el desarrollo iterativo.

Finalmente, se puede indicar que las metodologías ágiles tienen un modelo de proceso de desarrollo incremental e iterativo, y habitualmente están compuestas por un conjunto de principios, técnicas, artefactos y procedimientos que están directamente relacionados con los valores propuestos en el manifiesto ágil.

Una vez descritas y conocidas en detalle las metodologías ágiles, en la siguiente sección se aborda la viabilidad de integrar el Diseño Centrado en el Usuario en ellas, con el fin de identificar un marco de trabajo ideal para integrar la AI en entornos ágiles.

#### **2.3.4. Viabilidad de las Metodologías Ágiles para Integrar el Diseño Centrado en el Usuario**

Para llevar a cabo el estudio sobre la viabilidad de las metodologías ágiles e integrar el Diseño Centrado en el Usuario, se analizará cada una de las metodologías ágiles, utilizando un conjunto de atributos de referencia, basados en los principios de la ISO 9241-210 (DIS, 2009).

La ISO 9241-210 proporciona un marco para el Diseño Centrado en el Usuario. Ésta no asume ningún proceso de diseño en particular, ni tampoco describe todas las actividades necesarias para garantizar un diseño de sistemas eficaz, sino que se basa en los siguientes principios para promover el Diseño Centrado en el Usuario:

- *El Proceso Debe Ser Iterativo*: Este principio se refiere a que el diseño más apropiado para un sistema interactivo se debe lograr mediante iteraciones. Esto consiste en repetir una secuencia de pasos hasta que se consiga el resultado deseado.

- *Implicación Activa de Usuarios Finales*: Se trata de que los usuarios deben tener una participación activa, ya sea participando en el diseño, actuando como fuente de datos relevantes o evaluando las soluciones.
- *Equipos con Conocimiento Multidisciplinar*: El equipo de diseño debe incluir habilidades y perspectivas multidisciplinarias.
- *Evaluaciones Centradas en los Usuarios*: Este principio se refiere a que el diseño se debe impulsar y refinar por la evaluación centrada en el usuario.
- *Diseño Basado en los Usuarios*: Los productos se deben diseñar con una comprensión explícita de las personas que los utilizarán, así como de otros grupos de interesados, incluyendo a aquellos que podrían verse afectados (directa o indirectamente) por su uso.

Por lo tanto, se utilizarán estos principios como atributos de referencia para analizar la capacidad de las metodologías ágiles para integrar eficientemente el Diseño Centrado en el Usuario. En la Tabla 17, se presenta el análisis realizado a las metodologías ágiles. Como se puede observar, en la primera columna se identifican las metodologías ágiles analizadas, y en las otras columnas se especifican los principios de la ISO 9241-210. Cada principio se evalúa considerando los siguientes valores:

- *Si*: En los casos donde las propuestas cumplan completamente con el principio.
- *Parcialmente*: Para aquellas propuestas que cumplan el principio de manera parcial.
- *No*: En los casos donde las propuestas no cumplan con el principio.

Met. Ágiles	El Proceso Debe Ser Iterativo	Implicación Activa de Usuarios Finales	Equipos con Conocimiento Multidisciplinar	Evaluaciones Centradas en los Usuarios	Diseño Basado en los Usuarios
Scrum	<b>Si.</b> Se utiliza el <i>Sprint</i> para llevar a cabo el proyecto de forma iterativa.	<b>Parcialmente.</b> El rol <i>Dueño del Producto</i> representa a los intereses de los clientes y los usuarios, lo que permitiría involucrar parcialmente a los usuarios en el proceso.	<b>Si.</b> Los equipos Scrum son auto-organizados y multifuncionales, y se considera la participación de diferentes roles con diversas habilidades para llevar a cabo el proyecto.	<b>Parcialmente.</b> El rol <i>Dueño del Producto</i> (que representa los intereses de los usuarios) participa en las evaluaciones, lo que permitiría la participación de los usuarios en las evaluaciones. Sin embargo, las evaluaciones están centradas en la integración del sistema, lo que no garantizaría que estén centradas en el usuario.	<b>Parcialmente.</b> La lista de requisitos es organizada y priorizada por el <i>Dueño del Producto</i> , lo que permitiría la participación parcial de los intereses de los usuarios. Sin embargo, esto no garantiza que se efectúen estudios sobre las tareas y entornos que requieren los usuarios.
XP	<b>Si.</b> Se considera la ejecución de diferentes iteraciones para generar el producto final.	<b>Parcialmente.</b> Se contempla el uso de la práctica <i>Cliente en el Sitio</i> , donde un usuario real participa a tiempo completo como parte del equipo, lo que facilitaría la implicación activa de los usuarios en el proceso. No obstante,	<b>Parcialmente.</b> Se consideran varios roles, con el fin de cumplir con el proyecto. Sin embargo, los roles propuestos están orientados principalmente desde una perspectiva técnica y orientada al programador.	<b>Parcialmente.</b> Las evaluaciones están orientadas principalmente a los aspectos técnicos de programación. No obstante, el rol <i>Cliente</i> escribe pruebas para validar las características que están terminadas, lo que permitiría parcialmente centrar las evaluaciones en los usuarios.	<b>Parcialmente.</b> Se escriben <i>historias de usuarios</i> para reunir los requisitos que serán desarrollados por el equipo XP. Esto abre la oportunidad para incluir requisitos sobre los aspectos del entorno y las tareas de los usuarios. Sin embargo, las prácticas están orientadas a generar software,

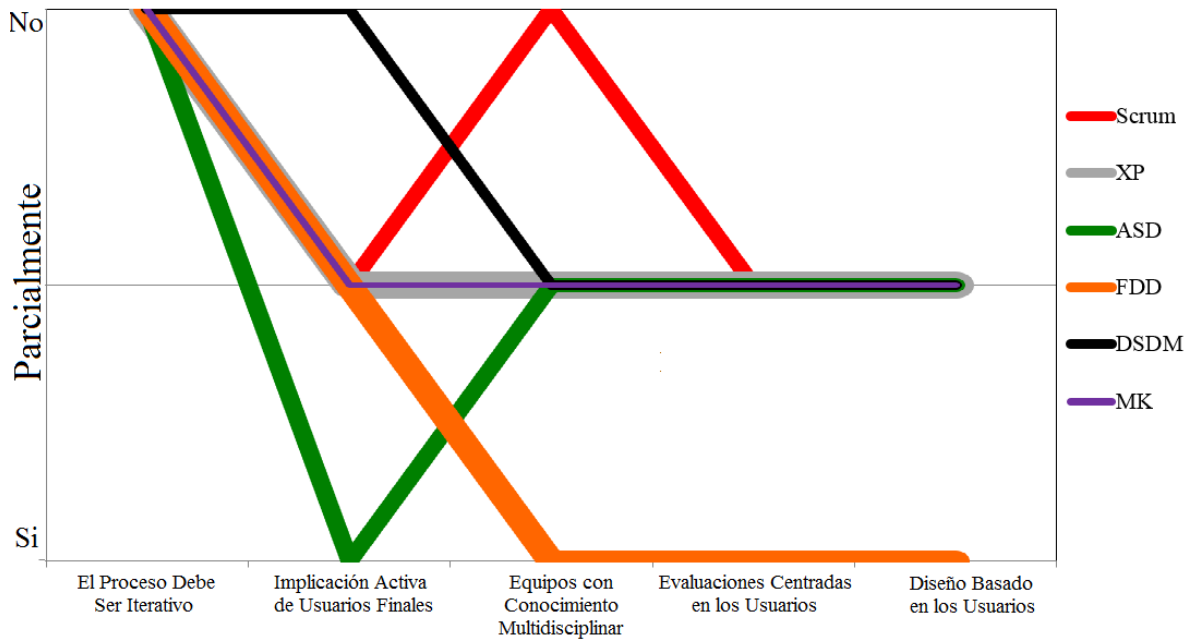
		esta práctica está orientada a promover la participación de los clientes del proyecto.			trabajando en desmedro de los aspectos relacionados con los usuarios finales.
ASD	<b>Si.</b> Se lleva a cabo el proyecto de forma iterativa.	<b>No.</b> Sólo se considera la participación de clientes para las evaluaciones del proyecto. Tampoco se describen roles relacionados con los usuarios para una posible participación activa en el proceso.	<b>Parcialmente.</b> No se describen estructuras de equipo en detalle. Sin embargo, se hace hincapié en la importancia de los equipos colaborativos y el trabajo en equipo, lo que supondría la conformación de un equipo con conocimiento multidisciplinar.	<b>Parcialmente.</b> Se contempla una etapa de revisión de calidad antes de la entrega final del proyecto, con uso de prácticas de evaluación centradas en los clientes, lo que supondría una oportunidad para integrar a los usuarios finales en las evaluaciones.	<b>Parcialmente.</b> El diseño se basa principalmente en la revisión de la calidad, respecto a los criterios de los clientes y los aspectos técnicos. Por lo tanto, los criterios de los clientes supondrían una oportunidad para incluir aspectos relacionados con las tareas y entornos de los usuarios.
FDD	<b>Si.</b> El proceso está dirigido por cortas iteraciones de cinco fases.	<b>Parcialmente.</b> El proceso está dirigido desde una perspectiva de la funcionalidad, que es valorada por el cliente. Esto parcialmente permitiría la activa participación del usuario.	<b>No.</b> La conformación del equipo de trabajo se centra, principalmente, en aspectos relacionados con el desarrollo y el mantenimiento de funcionalidades, desde una perspectiva técnica.	<b>No.</b> Aun cuando se menciona que las funcionalidades son llevadas a cabo desde una perspectiva valorada por el cliente, están dirigidas a aspectos técnicos de las funcionalidades construidas.	<b>No.</b> La construcción del proyecto está dirigida por las características y aspectos técnicos de las funcionalidades.

DSDM	<p><b>Si.</b> El proyecto se realiza mediante cinco fases, donde las dos primeras se llevan a cabo de forma secuencial y las tres últimas se realizan de forma iterativa e incremental.</p>	<p><b>Si.</b> Se utilizan prototipos como una de las técnicas clave del proyecto, con el fin de tener una activa participación de los usuarios en todo el proyecto. Asimismo, se indican roles relacionados con los usuarios.</p>	<p><b>Parcialmente.</b> Los roles propuestos están relacionados principalmente con los desarrolladores.</p>	<p><b>Parcialmente.</b> Se realizan revisiones de los prototipos con los usuarios. No obstante, el principal criterio de aceptación de un sistema es el propósito del negocio.</p>	<p><b>Parcialmente.</b> El uso de prototipos no funcionales y la participación de diferentes roles relacionados con los usuarios (<i>Embajador, Visionario y Asesor</i>) facilitaría la inclusión de aspectos sobre los entornos y las tareas de los usuarios. Sin embargo, se prioriza el hecho de conseguir una versión utilizable del software, por encima de reunir los aspectos necesarios de usabilidad.</p>
------	---	---	---	--	--

MK	<b>Si.</b> El flujo de trabajo es desarrollado de manera iterativa e incremental.	<b>Parcialmente.</b> Se permite adaptar los roles de acuerdo a las necesidades del proyecto. Esto facilitaría definir roles para una activa implicación de los usuarios finales. No obstante, cada etapa del flujo de trabajo se conduce por las prioridades de los clientes.	<b>Parcialmente.</b> No se establece ningún tipo de conformación de equipo de trabajo. No obstante, se permite incorporar roles adicionales, de acuerdo a las necesidades de cada equipo de trabajo. Esto permitiría incorporar roles con diferentes destrezas para realizar el proyecto.	<b>Parcialmente.</b> El enfoque considera la incorporación de ciclos de retroalimentación para evaluar el trabajo realizado, que supondría la posibilidad de centrar estas evaluaciones en los usuarios finales. No obstante, la prioridad está centrada en equilibrar la demanda y el rendimiento.	<b>Parcialmente.</b> El desarrollo del proyecto está centrado en generar productos de calidad, considerando las prioridades de los clientes. Por lo tanto, estos aspectos de calidad constituirían una oportunidad para relacionarlos con las características de las tareas y entornos de los usuarios.
----	---	---	---	---	---

**Tabla 17:** Análisis de la viabilidad de las metodologías ágiles para integrar el Diseño Centrado en el Usuario.

En la Figura 8, se presenta el resumen gráfico del análisis realizado a las metodologías ágiles. El eje X representa los atributos de la ISO 9241-210, utilizados para evaluar las metodologías ágiles, mientras que el eje Y indica el grado en que las metodologías ágiles cumplen con dichos atributos (Sí, Parcialmente o No).



**Figura 8:** Resumen gráfico de la evaluación de la viabilidad de las metodologías ágiles para integrar el Diseño Centrado en el Usuario.

Como se muestra en la Figura 8, todas las metodologías cumplen con el atributo *El Proceso Debe Ser Iterativo*, es decir, el proyecto se planifica y ejecuta de manera iterativa.

En cuanto al atributo de *Implicación Activa de Usuarios Finales*, en la mayoría de las metodologías ágiles (Scrum, XP, FDD, y MK) los usuarios finales están involucrados parcialmente en el proceso. La metodología ágil DSDM es la única que permite la participación activa de los usuarios, mediante la propuesta de roles que son asumidos directamente por ellos. Aunque algunas metodologías ágiles (Scrum y XP) también describen roles relacionados con los usuarios, éstos están más relacionados con los aspectos funcionales del sistema. También es importante destacar que Scrum incluye una revisión y mejora de los procesos, lo que facilitaría la participación de los usuarios finales.

Con respecto al atributo de *Equipos con Conocimiento Multidisciplinar*, la mayoría de las metodologías ágiles (Scrum, XP, ASD, DSDM y MK) consideran equipos de trabajo con diferentes habilidades y conocimientos. Sin embargo, la formación de equipos se basa principalmente en la búsqueda de la eficiencia en el desarrollo y mantenimiento de las funcionalidades, desde la perspectiva técnica de un programador de software. En el caso de la metodología Scrum, se señala expresamente que la formación de equipos de trabajo se basa en las habilidades necesarias para cumplir con el proyecto.



Con respecto al atributo de *Evaluaciones Centradas en los Usuarios*, la mayoría de las metodologías ágiles (Scrum, XP, ASD, DSDM y MK) realizan evaluaciones centradas parcialmente en los usuarios, considerándolos mediante el uso de roles relacionados con el cliente o usuarios expertos (con conocimiento de los procedimientos del sistema o de los requisitos funcionales), lo que supondría, en la práctica, la participación parcial de los usuarios finales en las evaluaciones del sistema. Sin embargo, estas evaluaciones están orientadas principalmente a pruebas de unidad e integración de los sistemas, ya que la mayoría de las metodologías ágiles se centran en el cliente y no en el usuario final. También es importante destacar que Scrum define un evento específico, con el fin de verificar cada uno de los incrementos del producto con la participación de las partes interesadas. Esta característica permitiría la participación de los usuarios finales en las evaluaciones del sistema.

En cuanto al último atributo de *Diseño Basado en los Usuarios*, en la mayoría de las metodologías ágiles (Scrum, XP, ASD, DSDM y MK), el diseño se basa en parte en la comprensión de los usuarios, tareas y entornos. En algunos casos (Scrum y DSDM), existe el uso de prototipos no funcionales y la participación de diversos roles relacionados (directa o parcialmente) con los usuarios. Esto facilita la inclusión de aspectos sobre los entornos y las tareas de los usuarios. Sin embargo, la prioridad está puesta en conseguir una versión utilizable del software. Un aspecto a destacar, es que en Scrum, los requisitos se gestionan de forma dinámica por el rol *Dueño del Producto* que tiene un perfil orientado a las necesidades del usuario. Por lo tanto, esto facilitaría el diseño basado en la comprensión de los usuarios, tareas y entornos.

Resumiendo, las metodologías ágiles que permiten una mejor integración del Diseño Centrado en el Usuario corresponden a DSDM y Scrum. Estas metodologías tienen en común la involucración de los usuarios finales a través de la consideración de roles específicos y el uso de prácticas. Por ejemplo, DSDM presenta descripciones de roles de usuario y del uso de prototipos. De una manera similar, Scrum faculta a los usuarios finales para poder priorizar los requisitos, así como también ofrece procesos de revisión y de mejora que facilitan la participación de los usuarios finales. No obstante, la metodología Scrum tiene algunas ventajas que facilitarían una hipotética integración ágil con el Diseño Centrado en el Usuario de una mejor forma. Por ejemplo, la metodología Scrum presenta un marco de trabajo que es fácil de implementar, proporcionando flexibilidad y adaptación a las necesidades requeridas por los usuarios finales y el negocio. Por el contrario, la metodología DSDM requiere de un marco institucional específico para el proceso de desarrollo de software, que no es ni barato ni fácil de implementar, y también exige un cambio cultural importante en la organización.

Por otra parte, las metodologías ASD y FDD son las que menos facilitan una integración centrada en los usuarios. Estas metodologías tienen en común su orientación hacia la búsqueda de técnicas de optimización de codificación y modelado para el desarrollo de

sistemas, así como el uso de métodos, enfocados en los aspectos técnicos del desarrollo de sistemas.

Por lo tanto, se puede indicar que la metodología ágil Scrum es una alternativa factible para utilizar como base para una integración de la AI, dentro de un proceso ágil de desarrollo centrado en el usuario. No obstante, las metodologías ágiles presentan algunos aspectos que dificultan integrar el Diseño Centrado en el Usuario. Es por esto que, en la siguiente sección, se analizarán las recomendaciones y prácticas para integrar eficientemente el Diseño Centrado en el Usuario en las metodologías ágiles, especialmente en la metodología Scrum.

Finalmente, todos estos resultados permiten corroborar la hipótesis de partida **H1.3**, al concluir que la mayoría de las metodologías ágiles presentan inconvenientes y dificultades para integrar el Diseño Centrado en el Usuario. Asimismo, estos resultados también permiten responder las preguntas de la investigación **PI<sub>7</sub>**, **PI<sub>8</sub>** y **PI<sub>9</sub>**, indicadas anteriormente en la sección 1.2:

- **PI<sub>7</sub>**: ¿Qué metodologías de desarrollo de software son comúnmente consideradas como ágiles?  
Las metodologías de desarrollo de software, que son indicadas generalmente como ágiles, corresponden a: Scrum, Programación Extrema, Desarrollo de Software Adaptativo, Desarrollo de Software Dirigido por Funcionalidades, Método de Desarrollo de Sistemas Dinámicos y el Método Kanban.
- **PI<sub>8</sub>**: ¿Qué enfoque de desarrollo y modelo de proceso son los más habituales en las metodologías ágiles?  
Los resultados del análisis y descripción de las metodologías ágiles muestran que éstas presentan un enfoque de desarrollo ágil y centrado en el cliente, y que tienen un modelo de proceso incremental e iterativo.
- **PI<sub>9</sub>**: ¿Qué metodología ágil permite una mejor integración del Diseño Centrado en el Usuario?  
La metodología ágil Scrum es la que presenta el marco de trabajo ágil más adecuado para integrar el Diseño Centrado en el Usuario.

## 2.4. Integración del DCU dentro del Proceso de Desarrollo Ágil

Esta sección se origina a partir del planteamiento de la hipótesis de partida **H1.4**: Las metodologías ágiles requieren de prácticas y recomendaciones específicas para integrar de manera eficiente el Diseño Centrado en el Usuario. Para la corroboración de esta hipótesis de partida, se estudian las técnicas, los artefactos y las recomendaciones, que usualmente se utilizan para integrar el Diseño Centrado en el Usuario dentro del proceso ágil de desarrollo. El objetivo de esta sección es identificar un conjunto de prácticas que puedan utilizarse como fuente de información importante para una propuesta concreta sobre la integración de la Arquitectura de la Información, mediante el marco de trabajo de Scrum.

Asimismo, con la realización de este estudio también se busca responder las siguientes preguntas de investigación (definidas en la sección 1.2):

**PI10:** ¿Qué técnicas se utilizan frecuentemente en la integración del Diseño Centrado en el Usuario dentro del paradigma ágil, y facilitan integrar la Arquitectura de la Información?

**PI11:** ¿Qué prácticas y recomendaciones específicas se utilizan para integrar el Diseño Centrado en el Usuario en la metodología Scrum?

### 2.4.1. Prácticas y Artefactos Comunes

En esta sección se revisan diferentes trabajos y estudios, que identifican las técnicas que frecuentemente se utilizan para integrar el Diseño Centrado en el Usuario dentro del paradigma ágil. También se examinan las técnicas de usabilidad, que normalmente se manipulan en Scrum, con el fin de identificar si existen correspondencias entre todas estas técnicas.

Silva et al. (Silva et al., 2011) identificaron, mediante una revisión sistemática de la bibliografía, cuáles son las necesidades, los artefactos y las prácticas comunes que se utilizan para apoyar la colaboración entre los diseñadores y desarrolladores en la integración del DCU, dentro del enfoque ágil. Silva et al. analizaron 58 documentos, concluyendo que las prácticas y los artefactos más importantes corresponden a: *Colaboración Cercana, Prototipos de Baja Fidelidad, Pruebas de Usuarios, Historias de los Usuarios, Métodos de Inspección y Panorama General*.

Del mismo modo, Jurca et al. (Jurca et al., 2014) realizaron un estudio sistemático de la literatura, para identificar las investigaciones relevantes, relacionadas con la combinación del DCU y el paradigma ágil. Algunas de las prácticas y artefactos identificados corresponden a: *Mapas Conceptuales, Prototipos de Baja Fidelidad, Entrevistas, Escenarios y Reuniones con los Usuarios*.

Recientemente, Brhel et al. (Brhel et al., 2015) también llevaron a cabo una revisión sistemática de la literatura sobre los aspectos de la integración del DCU en las metodologías ágiles. Los autores identificaron diversas prácticas mediante un sistema de codificación de cuatro niveles: procesos, prácticas, personas y dimensiones tecnológicas. De este modo, las prácticas más comunes identificadas corresponden a: *Prototipos, Escenarios, Evaluación de la Usabilidad (Experto), Historias de Usuarios y Pruebas de Usabilidad (Usuario)*, entre otras.

Finalmente, Jia et al. (Jia et al., 2012) condujeron un estudio para explorar cómo se han integrado las técnicas de usabilidad durante el desarrollo de software en proyectos Scrum. En este caso, las técnicas de usabilidad más utilizadas fueron: *Talleres, Prototipos de Baja Fidelidad, Entrevistas y Reuniones con los Usuarios*.

Así pues, se ha elaborado la Tabla 18, con el fin de identificar las correspondencias entre las técnicas, identificadas por (Silva et al., 2011; Jurca et al., 2014; Brhel et al., 2015), que

se utilizan para integrar el DCU dentro de las metodologías ágiles. Además, se identifican las correspondencias entre las técnicas mencionadas anteriormente y las técnicas de usabilidad, utilizadas en la metodología Scrum.

Técnicas Usadas para Integrar el DCU en las Metodologías Ágiles			Técnicas de Usabilidad Usadas en Scrum
(Silva et al., 2011)	(Jurca et al., 2014)	(Brhel et al., 2015)	(Jia et al., 2012)
Prototipo de Baja Fidelidad	Prototipo de Baja Fidelidad	Prototipado	Prototipo de Baja Fidelidad
Escenarios	Escenarios	Escenarios	Escenarios
Métodos de Inspección	Variantes de Recorrido Cognitivo	Evaluación de la Usabilidad (Experto)	Evaluación Heurística
Pruebas de Usuarios		Pruebas de Usabilidad (Usuarios)	Evaluaciones de Usabilidad Formal e Informal con Usuarios
Persona		Persona	Persona
	Talleres	Grupos de Discusión	Talleres
	Entrevistas	Entrevistas	Entrevistas
	Reuniones con Usuarios		Reuniones con Usuarios
		Investigación Contextual	Estudio de Campo
Historias de Usuario		Historias de Usuario	
Directrices		Directrices	

**Tabla 18:** Resumen, de acuerdo con la bibliografía revisada, que describe las correspondencias entre las técnicas utilizadas para integrar el DCU en metodologías ágiles y las técnicas de usabilidad utilizadas en la metodología Scrum.

Como se muestra en la Tabla 18, las técnicas más frecuentes, encontradas en la revisión sistemática de la literatura, corresponden a: *Prototipos de Baja Fidelidad*, *Escenarios*, *Evaluación Heurística*, *Pruebas de Usabilidad*, *Persona*, *Talleres* y *Entrevistas*. Esto consolida a este conjunto de técnicas como obligatorias, cuando se intenta integrar los aspectos del DCU dentro de los proyectos ágiles. Por otra parte, es importante señalar que las técnicas *Historias de usuarios* y *Directrices* no se mencionan en (Jia et al., 2012) como

prácticas comunes en proyectos Scrum, porque estas técnicas se relacionan principalmente con la metodología XP.

Por otra parte, aunque estas técnicas facilitan integrar la AI considerando las necesidades de los usuarios finales y respondiendo rápidamente en los entornos dinámicos, también se requiere estudiar las recomendaciones específicas que se utilizan para integrar el DCU dentro de la metodología Scrum, identificando correctamente el problema que resuelven y la solución que aportan.

### **2.4.2. Recomendaciones Específicas para Integrar el DCU en Scrum**

Una vez estudiadas las diferentes técnicas para integrar el DCU dentro de las metodologías ágiles, se recopilan y analizan las recomendaciones específicas, que se utilizan para integrar el DCU dentro de la metodología Scrum. A continuación, se presenta cada una de estas recomendaciones:

- *Definición de “Terminado” para las Tareas Relacionadas con el DCU:* Kniberg (Kniberg, 2007) puso de manifiesto la importancia de que el Dueño del Producto y el Equipo de Desarrollo deban ponerse de acuerdo sobre una definición clara de “Terminado”. Esto facilitaría obtener un entendimiento común sobre el alcance y las demandas de los requisitos, presentados por el Dueño del Producto, así como las tareas que se llevarán a cabo por el Equipo de Desarrollo. Felker et al. (Felker et al., 2012) han propuesto, como parte de un enfoque para la integración del DCU y Scrum, utilizar una definición diferente de “Terminado” para las tareas relacionadas con el diseño y la implementación. Estos autores indicaron que esta estrategia podría facilitar el seguimiento de los trabajos, con el fin de saber el momento adecuado para pasar a la siguiente tarea.
- *Panorama General del Proyecto:* El término *panorama general* se refiere a la visión general de todo el proyecto en Scrum. Lárusdóttir et al. (Lárusdóttir et al., 2012) indicaron que el panorama general, relacionado con la experiencia del usuario, suele faltar en los proyectos de Scrum. Para hacer frente a esta dificultad, Budwig et al. (Budwig et al., 2009) propusieron incorporar, trimestralmente, actividades orientadas a actualizar el panorama general, con el fin de tener una visión clara del diseño que se lleva a cabo en el proyecto, y mantener la coherencia global. Otra propuesta es utilizar los objetivos generales de calidad, para ayudar a entregar las directrices generales del diseño (Cajander et al., 2013).
- *Gestión de la Lista del Producto:* Singh (Singh, 2008) notó que uno de los principales desafíos para la usabilidad en los proyectos Scrum, es el estudio de las necesidades de los usuarios finales y el contexto. Además, este autor identificó que los requisitos seleccionados se deben ajustar a las exigencias de la usabilidad y, en consecuencia, priorizarlos. Para hacer frente a estos inconvenientes, Budwig et al. (Budwig et al., 2009) propusieron la creación de una Lista del Producto para temas relacionados con el DCU, que ayuda al equipo de DCU a asignar los recursos

correspondientes a los proyectos. Del mismo modo, Singh (Singh, 2008) propuso mantener la misma estructura de la Lista del Producto, pero incorporando elementos que incluyan una mayor conciencia de la usabilidad, es decir, una mayor prioridad para los requisitos que tienen un impacto en la usabilidad, especialmente en relación con los posibles criterios de aceptación de los requisitos. Por otra parte, Kuusinen (Kuusinen, 2014) ha propuesto que los especialistas del DCU tengan roles más influyentes, en lo que se refiere a las decisiones de nivel de producto, con el fin de mejorar la gestión del panorama general, y la comprensión y el cumplimiento de las necesidades de los usuarios finales.

- *Gestión de la Evaluación de la Usabilidad:* Lárusdóttir et al. (Lárusdóttir et al., 2012) indicaron que es difícil encontrar un buen momento para la evaluación del DCU en Scrum. Por un lado, una evaluación muy temprana en el proyecto es complicada debido a que las características disponibles son todavía insignificantes para tener una evaluación del DCU. Por otro lado, cuando las características son importantes para ser evaluadas, es difícil hacer cambios importantes, debido a que algunas de las partes del producto se han entregado y no hay tiempo suficiente para evaluar antes de la próxima entrega. Se han realizado diferentes propuestas para intentar minimizar este inconveniente. Entre éstas, Felker et al. (Felker et al., 2012) propusieron programar las evaluaciones antes de saber lo que se va a evaluar, analizar la retroalimentación del usuario final justo después de las evaluaciones del DCU, y llevarlas a cabo al final del Sprint. Similarmente, Lárusdóttir et al. (Lárusdóttir et al., 2012) recomendaron el uso de formas informales, para involucrar a los usuarios finales en la evaluación, y aplicar diferentes métodos para realizar con éxito cada una de las evaluaciones centradas en el usuario.
- *Completando la Investigación Contextual de Antemano:* La investigación contextual es una práctica, que se utiliza para inspeccionar y entender a los usuarios finales y a sus lugares de trabajo. Rannikko (Rannikko, 2011) recomendó que la investigación contextual complete antes de iniciar el desarrollo de software. Felker et al. (Felker et al., 2012) informaron sobre un uso exitoso de esta directriz, por lo que los autores observaron que la posesión de los resultados de una investigación contextual era muy útil, y esto les permitió centrarse en el diseño y la implementación, ayudando a establecer las prioridades iniciales. Asimismo, se ha recomendado (Moré, 2010) completar la investigación contextual para identificar las técnicas relacionadas con la usabilidad, que se puedan utilizar en cada iteración del proyecto.
- *Colaboración Cercana:* Lárusdóttir et al. (Lárusdóttir et al., 2012) sugirieron que los especialistas del DCU deben trabajar en estrecha colaboración con los desarrolladores en los equipos de Scrum. Por otra parte, Kuusinen (Kuusinen, 2014) indicó que era necesario identificar el momento adecuado para que los especialistas del DCU puedan trabajar. Estas cuestiones se han abordado de diferentes maneras,

entre ellas, se ha sugerido que el DCU debe ocurrir en vías paralelas a la implementación, configurar los equipos del DCU para que trabajen en uno o dos Sprints por delante del Equipo de Desarrollo (Budwig et al., 2009), y diseñar un Sprint por delante de la implementación (Felker et al.,; Lárusdóttir et al., 2014).

- *Asignar Responsabilidad por las Preocupaciones del Usuario Final:* Cajander et al. (Cajander et al., 2013) identificaron que la responsabilidad por la perspectiva del usuario final no está clara en los proyectos de Scrum, y la perspectiva del usuario final a menudo no se discute ni se describe en los proyectos. Cajander et al. (Cajander et al., 2013) propusieron reforzar el énfasis en la perspectiva del usuario final mediante la aclaración y la comunicación, de forma explícita, de la responsabilidad de trabajar a través de la usabilidad. Esto incluye tanto quién trabajará con la usabilidad como quién será el responsable de la calidad del producto final. Sin embargo, esta propuesta no resuelve lo que podría hacerse en el marco de trabajo de Scrum, donde no hay responsabilidades formales, relativas a los aspectos de calidad, tales como la seguridad, la privacidad y el rendimiento. Cajander et al. (Cajander et al., 2013) proporcionaron algunos ejemplos del apoyo organizativo necesario: un mandato suficiente, el apoyo de la gestión, la competencia organizacional, así como una posición adecuada en el equipo, para ser capaces de contribuir a una mejor usabilidad.
- *Sistematizar el Proceso de Inclusión del Usuario Final:* Cajander et al. (Cajander et al., 2013) indicaron que los procesos ágiles generalmente no son compatibles con la participación de los usuarios finales. Más bien, los usuarios finales están implicados de manera informal. A menudo esto se hace basándose en la iniciativa personal y el conocimiento de los miembros del equipo acerca de la perspectiva del usuario final, en lugar de planificarlo de manera sistemática en el proceso de Scrum. Cajander et al. (Cajander et al., 2013) sugirieron que podría ser útil sistematizar el proceso, mostrando la participación de los usuarios finales y la retroalimentación del diseño como actividades generales en el proceso de desarrollo.

A lo largo de toda esta sección (2.4), se han revisado y analizado diferentes prácticas y recomendaciones que facilitan y promueven la integración del DCU, dentro de las metodologías ágiles. De igual manera, se han identificado diversas correspondencias entre las técnicas utilizadas para integrar el DCU en metodologías ágiles y las técnicas de usabilidad utilizadas en la metodología Scrum. Asimismo, se han analizado en detalle las recomendaciones específicas, que se utilizan para integrar el DCU en la metodología Scrum.

De los análisis anteriores, se pueden obtener algunas directrices importantes que permiten fomentar el Diseño Centrado en el Usuario, dentro del marco de trabajo de la metodología Scrum. Esto se debe a que se identifican recomendaciones que resuelven varias de las limitaciones, identificadas en la sección 2.3.4. Por lo tanto, se considerarán algunas de estas recomendaciones (por ejemplo, formalizar la responsabilidad por los temas relacionados

con la usabilidad, generar un panorama general de la experiencia del usuario, utilizar maneras informales para involucrar a los usuarios finales en las evaluaciones, entre otras) para diseñar y vislumbrar las características principales, para integrar la AI dentro de un proceso ágil de desarrollo centrado en el usuario. No obstante, también se debe asegurar que los requisitos del proyecto se completen de manera ágil, considerando las prioridades de la AI, de la usabilidad y del valor de negocio. Es por esto que, en la siguiente sección se estudiarán los métodos de priorización, con el objetivo de verificar si permiten capturar adecuadamente las prioridades de los proyectos y conducir el proceso de priorización de acuerdo a dichas prioridades.

Finalmente, estos resultados permiten corroborar la hipótesis de partida **H1.4**, al concluir que las metodologías ágiles demandan diferentes tipos de prácticas para integrar correctamente el Diseño Centrado en el Usuario. Asimismo, es posible responder las preguntas de investigación, indicadas anteriormente:

- **PI<sub>10</sub>**: ¿Qué técnicas se utilizan frecuentemente en la integración del Diseño Centrado en el Usuario dentro del paradigma ágil, y facilitan integrar la Arquitectura de la Información?

De acuerdo con los resultados de la sección 2.4.1, las técnicas que se utilizan comúnmente en la integración del Diseño Centrado en el Usuario dentro del paradigma ágil corresponden a: Prototipos de Baja Fidelidad, Escenarios, Evaluación Heurística, Pruebas de Usabilidad, Persona, Talleres y Entrevistas.

- **PI<sub>11</sub>**: ¿Qué prácticas y recomendaciones específicas se utilizan para integrar el Diseño Centrado en el Usuario en la metodología Scrum?

De acuerdo con los resultados del análisis bibliográfico sobre las recomendaciones para integrar el DCU en Scrum, las prácticas más comunes corresponden a:

- Generar una definición de “Terminado” para las tareas relacionadas con el DCU.
- Incorporar elementos en la Lista del Producto que incluyan una mayor conciencia de la usabilidad.
- Utilizar formas informales para involucrar a los usuarios finales en la evaluación.
- Realizar una investigación contextual, antes de iniciar el desarrollo de software.
- Establecer una estrecha colaboración entre los especialistas del DCU y los desarrolladores en los equipos de Scrum.
- Proveer un panorama general del proyecto, relacionado con la experiencia del usuario.
- Asignar la responsabilidad explícita de la perspectiva del usuario final.
- Sistematizar el proceso de la participación de los usuarios finales.



## 2.5. Priorización de Requisitos

Esta sección se origina a partir del planteamiento de la hipótesis de partida **H1.5**: Los métodos de priorización de requisitos actuales no permiten conducir de manera ágil el proceso de priorización y capturar formalmente las preferencias de la usabilidad, de la Arquitectura de la Información y del valor de negocio que definen la prioridad de los requisitos. Para la corroboración de esta hipótesis de partida, se revisarán las características y las limitaciones principales de los métodos de priorización actuales, con el fin de estudiar su capacidad para adaptarse a entornos dinámicos y concentrarse en los requisitos más importantes para integrar la Arquitectura de la Información.

Asimismo, con la revisión de los métodos de priorización, se busca responder las siguientes preguntas de investigación (definidas en la sección 1.2):

**PI<sub>12</sub>**: ¿Cuál es la importancia de tratar las *colisiones de requisitos* en los entornos ágiles?

**PI<sub>13</sub>**: ¿Qué limitaciones de los métodos de priorización actuales dificultan integrar la Arquitectura de la Información dentro de un marco ágil de desarrollo centrado en el usuario?

**PI<sub>14</sub>**: ¿Permiten los métodos de priorización actuales adaptarse a entornos ágiles y formalizar la evaluación de la priorización de los requisitos mediante el uso de elementos cualitativos relacionados con las prioridades del proyecto?

### 2.5.1. Definición

La priorización de los requisitos es una actividad fundamental en la ingeniería de requisitos. El objetivo de esta actividad es obtener los requisitos más esenciales, satisfaciendo las necesidades de las partes interesadas y cumpliendo las limitaciones presupuestarias y de planificación (Pohl y Rupp, 2011).

A pesar de que en la actualidad existan diferentes métodos de priorización que sirven para seleccionar y clasificar los requisitos relevantes, todavía hay retos e inconvenientes que deben afrontarse. Por un lado, Babar et al. (Babar et al., 2011) señalaron la importancia del punto de vista de las partes interesadas (esto implica considerar los requisitos desde el punto de vista de las partes interesadas), así como de la dependencia y de la escalabilidad (la forma en la que un método se comporta cuando aumenta el número de requisitos). Por otro lado, el principal inconveniente de los enfoques de priorización existentes es su énfasis en medidas cuantitativas (Achimugu et al., 2014), dificultando capturar eficientemente las prioridades de la usabilidad, de la AI y del valor de negocio, requeridas para discernir el nivel de relevancia de los requisitos.

Asimismo, la mayoría de los métodos de priorización existentes no consideran las *colisiones de requisitos* (Berander et al., 2006). Esto se refiere a la situación en la que dos o más requisitos tienen el mismo valor de priorización en un ranking, siendo éste un factor importante en el proceso ágil de desarrollo, donde los requisitos deben seleccionarse de

manera única para cada incremento del software. De esta manera, un adecuado tratamiento de las colisiones de requisitos en los entornos ágiles permite discernir el nivel de relevancia en los conjuntos de requisitos, mejorar la priorización en la toma de decisiones, y verificar la consistencia y exactitud de las evaluaciones en la clasificación mientras el número de requisitos aumenta.

Por lo tanto, en la siguiente sección se estudiarán las principales características y limitaciones de los métodos de priorización actuales, con el objetivo de identificar si estos permiten capturar de manera ágil las prioridades de la usabilidad, de la AI y del valor de negocio, y tratar adecuadamente las colisiones de requisitos, para integrar la AI dentro de un marco ágil de desarrollo centrado en el usuario de manera eficiente.

### 2.5.2. Métodos de Priorización

Actualmente, existe una gran variedad de métodos de priorización, utilizados en el proceso de desarrollo de software tradicional y ágil (Achimugu et al., 2014; Racheva et al., 2008; Racheva et al., 2010). Entre estos, el método *Proceso Analítico Jerárquico* (AHP<sup>6</sup>) (Saaty, 1980) es uno de los más populares y citados (Achimugu et al., 2014). AHP se utiliza en el campo de los requisitos de priorización para identificar la prioridad de cada requisito a través de una matriz de comparación por pares. El uso de AHP está muy extendido, debido a su facilidad de aplicación y estructura, así como su forma intuitiva de computar (Ishizaka y Labib, 2009). Ésta es una de las razones por las que hay varios métodos de priorización basados en AHP.

Por ejemplo, el método *Coste-Valor* (Karlsson y Ryan, 1997) utiliza AHP para priorizar los requisitos, en función de su valor percibido y el coste de implementación. Del mismo modo, el método *Análisis por Pares* (Karlsson et al., 1996) se basa en AHP para priorizar los requisitos mediante la comparación por pares, con el fin de determinar qué requisito se debe seleccionar. El método *Ranking Basado en Casos* (CBRank<sup>7</sup>) (Perini et al., 2013) también está influenciado por AHP, pero éste utiliza una técnica de aprendizaje automática, que reduce el esfuerzo humano en la entrada de la información requerida, manteniendo la exactitud de la clasificación final.

De manera similar, los métodos *AHP Difuso* (Lima et al., 2011) y *AHP Jerárquico* (Karlsson et al., 1998) utilizan AHP y técnicas de aprendizaje automáticas, con el fin de reducir el número requerido de comparación por pares. Por un lado, el método AHP Difuso lleva a cabo la priorización a través de objetivos poco claros, ponderando la clasificación de los requisitos mediante datos ambiguos y vagos. Por otro lado, en el método AHP *Jerárquico*, los interesados proponen un conjunto fijo de requisitos, con el fin de priorizarlos, desde el más general al más específico.

---

<sup>6</sup> Por sus siglas del inglés: **A**nalytic **H**ierarchy **P**rocess.

<sup>7</sup> Por sus siglas del inglés: **C**ase-**B**ased **R**anking.

Asimismo, el método *Algoritmo Genético Interactivo* (IGA<sup>8</sup>) (Tonella et al., 2013) también se basa en AHP, ofreciendo, sin embargo, un algoritmo genético para reducir el número de pares elicitados, y para obtener así el conocimiento del usuario sobre el nivel de relevancia para cada par de requisitos. Por último, el método *Conducción Cognitiva* (Carod y Cechich, 2010) combina la utilización de AHP y la psicología cognitiva para evaluar la capacidad de las partes interesadas, en relación con el software sugerido, antes de que comience el proceso de priorización de los requisitos.

No obstante, una de las limitaciones más comunes de los métodos mencionados anteriormente (AHP, Coste-Valor, Análisis por Pares, CBRank, AHP Difuso, AHP Jerárquico, IGA y Conducción Cognitiva) es que no afrontan las colisiones de requisitos durante todo el proceso de priorización. De hecho, estos métodos requieren de mucho tiempo cuando hay un gran número de requisitos, y son difícilmente escalables a largo plazo (Karlsson et al., 1998). Además, la mayoría de estos métodos cuentan con mecanismos de priorización, basados en la evaluación individual de los requisitos a través de comparaciones por pares. Esto carece de una comprensión global de los aspectos relevantes del proyecto, con el fin de guiar de manera dinámica el proceso de priorización, proporcionar los elementos cualitativos que permitan considerar las preferencias de la usabilidad, del valor de negocio y de la AI, y reducir las colisiones de los requisitos.

Continuando con el análisis de los enfoques de priorización, el método *Despliegue de la Función Calidad* (QFD<sup>9</sup>) (Crow, 1994) es el segundo método de priorización más citado (Achimugu et al., 2014). Este método utiliza una matriz para representar las necesidades y expectativas de las partes interesadas. Otro enfoque similar es el método *Evaluación de Prioridades Basada en la Correlación* (CBPA<sup>10</sup>) (Liu et al., 2006) que utiliza una matriz de relaciones para priorizar los requisitos provenientes de múltiples partes interesadas. El método *Kano* (Kano et al., 1984) también permite priorizar los requisitos basados en las preferencias de las partes interesadas, pero se centra en las diferencias de las características del producto. Del mismo modo, el método *Teoría de Lanchester* (Fehlmann, 2008) utiliza un modelo cuantitativo para impulsar la priorización de requisitos, donde éstos se priorizan de acuerdo a los objetivos del negocio y la cuota de mercado.

Similarmente, el método Weiger (Wieggers, 1999) estima las prioridades relativas de los requisitos a través de un esquema basado en el concepto de QFD sobre valor del cliente. Por último, el método *Definición de Producto* (Fraser, 2002) prioriza los requisitos teniendo en cuenta la perspectiva de los usuarios, la tecnología y los negocios, así como también fomentando la participación de las partes interesadas, los expertos en experiencia del usuario y los analistas técnicos.

---

<sup>8</sup> Por sus siglas del inglés: **I**nteractive **G**enetic **A**lgorithm.

<sup>9</sup> Por sus siglas del inglés: **Q**uality **F**unctional **D**eployment.

<sup>10</sup> Por sus siglas del inglés: **C**orrelation-**B**ased **P**riority **A**ssessment.

En general, los métodos de priorización antes mencionados (QFD, CBPA, Kano, Teoría de Lanchester, Weiger y Definición de Producto) permiten capturar diversos elementos para obtener una concepción global del proyecto (objetivos de negocio, las expectativas del cliente y necesidades de los interesados), y así conducir la priorización de los requisitos. Sin embargo, estos métodos son, en su mayoría, adecuados para pequeños conjuntos de requisitos, debido a que no tienen un buen comportamiento en términos de escalabilidad (Avesani et al., 2005). Del mismo modo, también tienen limitaciones con respecto al uso subjetivo de escalas ordinales (Achimugu et al., 2014). Esto hace que sea difícil proporcionar elementos cualitativos objetivos, con el fin de aclarar las inconsistencias, las necesidades de los usuarios finales y las colisiones de requisitos durante un proceso dinámico de priorización.

Por otro lado, el método *Árbol Binario de Búsqueda* (BST<sup>11</sup>) (Hopcroft, 1983; Karlsson et al., 1998) ofrece un árbol binario en el que los nodos están etiquetados con los requisitos en un orden jerárquico (relación padre-hijo). Asimismo, el método *Árbol-B* (Beg et al., 2009) también permite organizar los requisitos en nodos y compararlos para establecer el nivel de relevancia utilizando una escala ponderada, lo que reduce el número de comparaciones.

Sin embargo, los métodos de priorización anteriores (BST y *Árbol-B*) presentan algunas limitaciones en cuanto a la escalabilidad (Aasem et al., 2010) y la ausencia de valores de prioridad para la clasificación final de los requisitos (Duan et al., 2009). Además, las evaluaciones obtenidas en la priorización de requisitos no están basadas ni en medidas cualitativas ni relacionadas con los aspectos relevantes que definen las prioridades del proyecto, lo que dificulta analizar las preferencias de los usuarios finales, las colisiones de requisitos y la precisión de la clasificación final de manera ágil.

También es posible identificar los métodos de priorización utilizados específicamente en las metodologías ágiles, tales como *Juego de Planificación*, *MoSCoW*<sup>12</sup>, *Asignación \$100*, *Punto de Votación*, *Orientado al Valor*, *Sistema de Votación Múltiple* y *Análisis de Criterios Ponderados* (Racheva et al., 2008; Racheva et al., 2010). Para centrarse en los más importantes, se puede mencionar que el método *MoSCoW* (Stapleton, 1997) prioriza los requisitos basados en la identificación de aquellos con un valor más alto para el sistema, mientras que el método *Orientado al Valor* (Azar et al., 2007) evalúa los requisitos de acuerdo a los valores centrales del negocio y de las partes interesadas. Estos métodos de priorización son orientados a pequeños conjuntos de requisitos, lo que facilita su aplicación en los equipos de desarrollo dinámicos. No obstante, en la mayoría de los casos, estos métodos de priorización se centran, principalmente, en clasificar los requisitos, sin ninguna gestión de las colisiones e ignorando las dependencias entre los requisitos. Además, no

<sup>11</sup> Por sus siglas del inglés: **B**inary **S**earch **T**ree.

<sup>12</sup> Por sus siglas del inglés: **M**ust have, **S**hould have, **C**ould have, and **W**ould like but won't get.

tienen un buen comportamiento en términos de escalabilidad, mientras el número de requisitos crece (Aasem et al., 2010; Hatton, 2008).

En síntesis, es posible afirmar que los métodos de priorización existentes no tienen en cuenta a las colisiones de requisitos, con el fin de validar y mejorar la precisión de la clasificación. Además, las prioridades del proyecto son ignoradas o no son capturadas formalmente para verificar la consistencia de las principales preocupaciones relacionadas con los requisitos. Por último, la mayoría de los enfoques existentes se basan en medidas cuantitativas, dificultando la presentación de un buen comportamiento en términos de escalabilidad en entornos de gran escala y con gestión dinámica de requisitos.

Tales inconvenientes y limitaciones dificultan gestionar los requisitos de usabilidad de contenido en entornos cambiantes, y considerar la perspectiva de los usuarios finales durante todo el modelo de proceso. Por lo consiguiente, en el siguiente capítulo se presentará y describirá una propuesta concreta, para dar solución a los problemas encontrados en este estudio, con el fin de proveer una eficiente integración de la AI dentro de un proceso ágil de desarrollo centrado en el usuario.

Finalmente, los resultados de este estudio permiten corroborar la hipótesis de partida **H1.5**, al concluir que los métodos de priorización actuales no logran capturar formalmente las preferencias del proyecto y conducir de manera ágil el proceso de priorización. Asimismo, es factible responder las preguntas de investigación mencionadas anteriormente:

- **PI12:** ¿Cuál es la importancia de tratar las colisiones de requisitos en los entornos ágiles?

Es importante tratar las colisiones de requisitos en los entornos ágiles porque un número elevado de colisiones dificulta establecer las diferencias entre las prioridades de los requisitos, afectando de manera negativa la toma de decisiones en el proceso de priorización. De este modo, un tratamiento adecuado del número de colisiones de requisitos (es decir, la generación de un número bajo de colisiones de requisitos) permite obtener una clasificación final discernible y precisa.

- **PI13:** ¿Qué limitaciones de los métodos de priorización actuales dificultan integrar la Arquitectura de la Información dentro de un marco ágil de desarrollo centrado en el usuario?

Las principales limitaciones de los métodos de priorización que obstaculizan integrar correctamente la AI en un modelo ágil de desarrollo centrado en el usuario corresponden a:

- Las colisiones de requisitos son ignoradas o desestimadas en el proceso de priorización, lo que genera dificultades para verificar, de manera dinámica, la exactitud de la clasificación final de los requisitos.
- Las prioridades de la usabilidad, de la AI y del valor de negocio no son capturadas de manera formal, dificultando la comprobación de la consistencia de las principales preferencias de los requisitos.

- En la mayoría de los casos, el proceso de priorización de los requisitos se basa en medidas cuantitativas, dificultando la presentación de un buen comportamiento en términos de escalabilidad cuando el número de requisitos aumenta.
- **PI<sub>14</sub>:** ¿Permiten los métodos de priorización actuales adaptarse a entornos ágiles y formalizar la evaluación de la priorización de los requisitos mediante el uso de elementos cualitativos relacionados con las prioridades del proyecto?

La mayoría de los métodos de priorización se basan en medidas cuantitativas y en una evaluación individual de los requisitos, mediante comparaciones por pares, sin una comprensión global de las preferencias de los proyectos. Por lo tanto, es posible afirmar que los métodos de priorización actuales no capturan, de manera formal, las prioridades del proyecto a través de elementos cualitativos, dificultando su implementación en ambientes dinámicos y complejos.

## 2.6. Recapitulación

El presente capítulo sobre el estado del arte ha permitido, a través de un estudio sistemático de la literatura, señalar que no existen trabajos anteriores que realicen contribuciones específicas para abordar una integración de la AI dentro de un proceso ágil de desarrollo centrado en el usuario. Esto ha permitido corroborar la hipótesis de partida **H1.1**.

Asimismo, una vez que se analizaron en profundidad las diferentes características de las propuestas actuales para la AI, fue factible determinar que estas propuestas no presentan un modelo de proceso ágil de desarrollo centrado en el usuario, lo que ha permitido corroborar la hipótesis de partida **H1.2**. Adicionalmente, después de analizar en detalle las diferentes metodologías ágiles, fue posible discernir claramente que la metodología Scrum presenta un marco de trabajo que facilita integrar la AI dentro de un modelo ágil de desarrollo centrado en el usuario, lo que ha permitido corroborar la hipótesis de partida **H1.3**. De igual manera, fue posible identificar las prácticas que se utilizan comúnmente para incorporar el Diseño Centrado en el Usuario dentro del proceso ágil de desarrollo, y que facilitan integrar la AI, así como las recomendaciones específicas empleadas en la metodología Scrum. Esto ha permitido corroborar la hipótesis de partida **H1.4**. Además, después de examinar los métodos de priorización actuales, fue posible identificar sus inconvenientes que obstaculizan priorizar, de manera dinámica, los requisitos de usabilidad de contenido, y capturar las principales prioridades de la usabilidad, de la AI y del valor de negocio, lo que ha llevado a la corroboración de la hipótesis de partida **H1.5**.

En resumen, la corroboración de las hipótesis de partida anteriores permite justificar la elaboración de esta tesis doctoral y obtener las directrices principales para realizar una contribución concreta sobre una integración ágil de la AI, donde sus elementos se desarrollen de manera iterativa e incremental, conducida por las prioridades y necesidades de los usuarios finales.

# Capítulo 3: Una Propuesta para Integrar la Arquitectura de la Información en un Proceso Ágil de Desarrollo Centrado en el Usuario

---

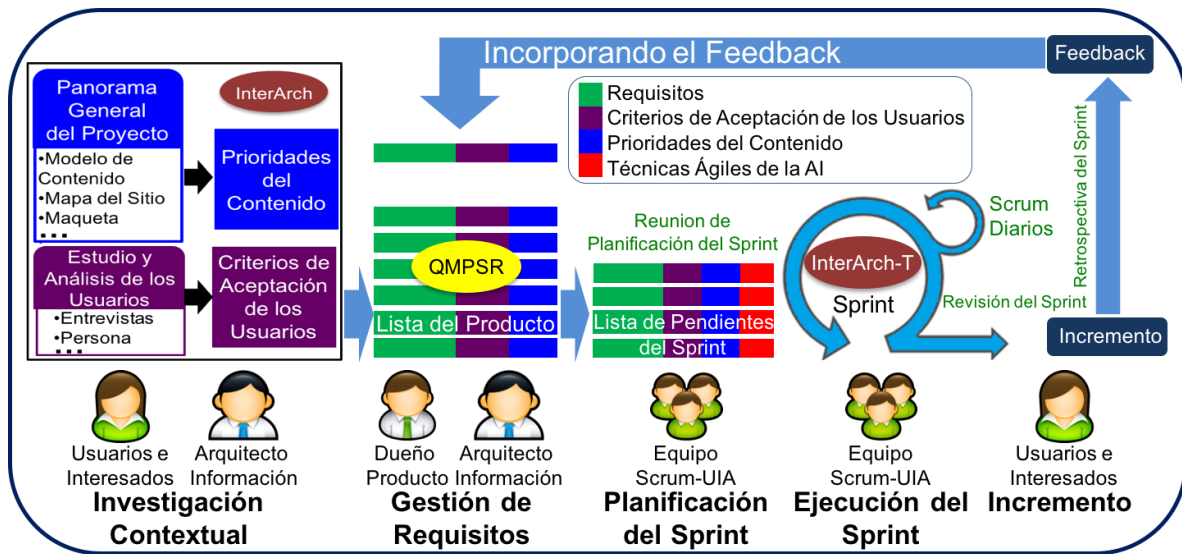
**E**n este capítulo, se describe la solución propuesta para integrar la Arquitectura de la Información dentro de un proceso ágil de desarrollo centrado en el usuario que permite adaptar las actividades de la AI para el desarrollo de software, incrementando así la satisfacción de los usuarios finales.

En primer lugar, en la sección 3.1, se introducen los lineamientos generales de la solución propuesta. Posteriormente, en la sección 3.2, se presentan la descripción del modelo de proceso de Scrum-UIA y la prescripción de todas las actividades propuestas, junto con los productos generados en detalle. En la sección 3.3, se describe el método de priorización de requisitos de la metodología Scrum-UIA, mientras que en la sección 3.4, se especifican la técnica propuesta para conducir el desarrollo incremental de la metodología Scrum-UIA, así como su implementación a través de una herramienta CASE. Finalmente, en la sección 0, se presenta la herramienta que sistematiza la aplicación de la metodología Scrum-UIA.

## 3.1. Descripción General de la Propuesta

Con el fin de hacer frente a los inconvenientes comentados anteriormente, y considerando también las prácticas analizadas para integrar el DCU en el paradigma ágil, se propone una metodología, llamada Scrum-UIA (**Scrum** driven by **Usable Information Architecture**) (Rojas y Macías, 2015), para integrar la AI a través de un proceso ágil de desarrollo centrado en el usuario. Esta metodología se basa en el marco de trabajo de Scrum e incluye roles, eventos, artefactos y reglas asociadas. Este enfoque es una combinación de las recomendaciones y de las prácticas específicas para integrar el DCU en el proceso de Scrum (ver Figura 9) que permite implementar los requisitos a través de un modelo de desarrollo incremental e iterativo.

Como se muestra en la Figura 9, el equipo Scrum se compone siguiendo la misma estructura tradicional de Scrum (un Dueño del Producto, el Equipo de Desarrollo y un Scrum-UIA Master), pero se incorpora al Arquitecto de la Información con un rol primario, con el fin de dirigir la Investigación Contextual, apoyar la gestión de la Lista del Producto, promover el desarrollo incremental, dirigido por la AI, asegurar el DCU y fomentar la participación de los usuarios finales.



**Figura 9:** Detalles de Scrum-UIA.

Scrum-UIA se basa en el establecimiento de una Investigación Contextual como el punto de partida del proyecto, con el objetivo de estudiar y analizar las necesidades de los usuarios finales, así como priorizar los contenidos. De esta forma, se adquieren los conocimientos y se proporcionan las bases para establecer un panorama general del proyecto. En este contexto, el Arquitecto de la Información crea modelos de contenido, relacionados con las prioridades de los contenidos del proyecto, a través del uso de la herramienta InterArch (ver InterArch que aparece en la Figura 9 durante la Investigación Contextual). Esto permite proveer continuidad entre la información de análisis conceptual de la AI y la información de análisis y diseño, que posteriormente necesitan los miembros del Equipo de Desarrollo en actividades relacionadas con el diseño de la aplicación.

Posteriormente, en la Gestión de los Requisitos, los requisitos que son incluidos en la Lista del Producto se priorizan de manera dinámica, de acuerdo a las prioridades obtenidas en la Investigación Contextual. Este proceso de priorización se conduce de acuerdo al enfoque del método de priorización propuesto (ver QMPSR, en la Figura 9), que permite considerar, de manera formal, las prioridades relacionadas con la usabilidad, la AI y el valor de negocio.

Tras la Gestión de Requisitos, se realiza la Planificación del Sprint, mediante el evento Reunión de Planificación del Sprint. En este evento se define el trabajo a realizar en cada Sprint del proyecto. Por lo tanto, se seleccionan los requisitos de mayor prioridad de la Lista del Producto para la Lista de Pendientes del Sprint. Tal lista se compone de los requisitos, seleccionados de la Lista del Producto, junto con un plan para llevarlos a cabo. Posteriormente, el Equipo de Desarrollo divide dichos requisitos en tareas específicas, que



se desarrollan durante la Ejecución del Sprint. Cabe destacar que la ejecución de estas tareas se organiza para promover el desarrollo incremental en Scrum-UIA a través de la AI.

Asimismo, en las tareas específicas de desarrollo, se pueden asociar diferentes técnicas para el desarrollo ágil de la AI centrada en los usuarios, que son auto-asignadas por el Equipo de Desarrollo. Es importante mencionar, que la técnica InterArch-T también se puede utilizar para elaborar las tareas específicas de desarrollo, relacionadas con la AI (ver InterArch-T, que aparece en la Figura 9 durante la Ejecución del Sprint).

Seguidamente, durante la Ejecución del Sprint, se llevan a cabo las reuniones de los Scrum Diarios. Estas reuniones se realizan para inspeccionar y adaptar el trabajo diario, efectuado por el Equipo de Desarrollo, y también para gestionar las tareas de desarrollo individuales, con el fin de mantener la visibilidad y la coherencia con el panorama general.

La reunión de la Revisión del Sprint se lleva a cabo al final de la Ejecución del Sprint, con la participación de los usuarios finales y otras partes secundarias interesadas. El objetivo de esta reunión es analizar el incremento potencial del producto, generado por el Equipo de Desarrollo.

Por último, se realiza la reunión de la Retrospectiva del Sprint, con fin de que el equipo Scrum-UIA pueda analizar las prácticas de trabajo, utilizadas durante la iteración, y buscar oportunidades de mejora.

En general, Scrum-UIA se basa en tres directrices esenciales que permiten integrar la AI dentro de un marco ágil de desarrollo centrado en el usuario:

- Gestión de la Lista del Producto Dirigida por la Investigación Contextual.
- Planificación del Sprint Dirigida por la Arquitectura de la Información.
- Procesos de Inspección y Mejora Continua Dirigidos por el Usuario Final.

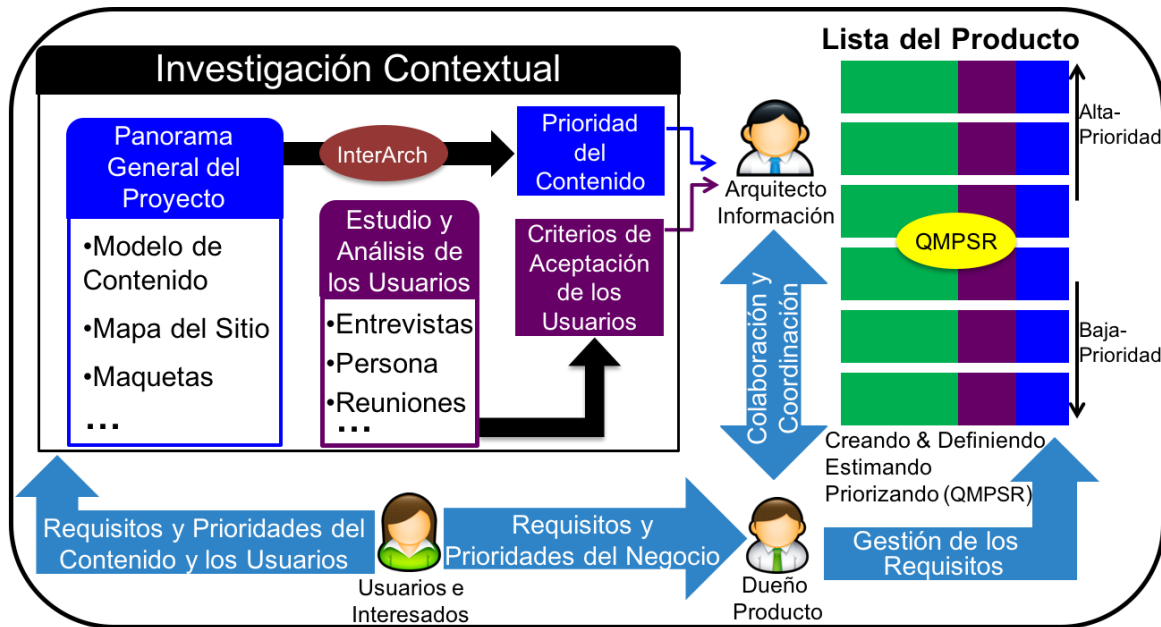
A continuación, se describen estas directrices esenciales.

### **3.1.1. Gestión de la Lista del Producto Dirigida por la Investigación Contextual**

El objetivo de esta directriz es doble. En primer lugar, se trata de asegurar que la perspectiva del usuario final se pueda discutir, describir y considerar durante todo el proceso de Scrum-UIA, problema reportado por (Cajander et al., 2013). En segundo lugar, se pretende proporcionar un panorama general del proyecto para ayudar a obtener una visión global e integradora del producto con respecto a las prioridades de la usabilidad, de la AI y del valor de negocio.

Con el fin de cumplir con los objetivos indicados anteriormente, se propone iniciar el proyecto con una Investigación Contextual (ver Figura 10), como se recomienda en (Rannikko, 2011) y se implementó con éxito en (Felker et al., 2012), con el objetivo de obtener conocimientos acerca de las prioridades de los usuarios finales y del contenido.

Esto proporciona, al mismo tiempo, las bases para configurar el panorama general del proyecto, para apoyar la gestión de la Lista del Producto.



**Figura 10:** Gestión de la Lista del Producto Dirigida por la Investigación Contextual.

Lárusdóttir et al. (Lárusdóttir et al., 2012) indicaron que los proyectos Scrum no tienen un panorama general y, de acuerdo con (Silva, 2012), esto se percibe como un problema entre los profesionales del DCU. Por lo tanto, se propone abordar este problema mediante la especificación de una visión de baja fidelidad de los siguientes artefactos: Modelos de Contenido, Planos y Maquetas. Estos artefactos permiten informar sobre el diseño de la navegación contextual de un sitio Web, determinar el contenido crítico y visualizar las relaciones entre las páginas y otros componentes de contenido. Por lo tanto, estos artefactos se pueden procesar para obtener un panorama general de cada proyecto y, de esta forma, facilitar la discusión sobre la organización y la gestión de los contenidos, así como sobre las prioridades de acceso deseadas por los usuarios finales (Morville y Rosenfeld, 2006; Brown, 2010).

Como resultado, los artefactos sugeridos permiten apoyar la gestión del panorama general del proyecto, proporcionando una visión clara de los elementos de contenido más críticos de un sitio. Además, éstos también facilitan el debate entre los miembros del equipo sobre los aspectos de los usuarios finales y de los contenidos, relacionados con las necesidades del negocio.

Por lo tanto, se propone desarrollar los artefactos sugeridos antes de iniciar el proyecto actual (Lárusdóttir et al., 2012), es decir, durante la Investigación Contextual. Por otra parte, se sugiere incorporar trabajos trimestrales para actualizar los artefactos (Budwig et

al., 2009) y utilizar los objetivos generales de calidad para proveer las directrices generales del diseño (Cajander et al., 2013). Asimismo, se sugiere obtener y especificar los criterios de aceptación de los usuarios finales a través de las técnicas *Entrevistas*, *Persona* y *Reuniones* que se utilizan habitualmente en los proyectos de Scrum (Jia et al., 2012).

Por otro lado, como se muestra en la Figura 10, se propone, de manera similar a la sugerencia de Kuusinen (Kuusinen, 2014), proveer al Arquitecto de la Información con la facilidad para definir, estimar y priorizar la Lista del Producto, en colaboración y coordinación con el Dueño del Producto. En este caso, el Arquitecto de la Información proporciona la visión de la AI y de la usabilidad, obtenida desde la Investigación Contextual, mientras que el Dueño del Producto proporciona los requisitos y las prioridades del negocio.

Finalmente, de una manera parecida a (Singh, 2008), se propone mantener la misma estructura de la Lista del Producto, incorporando, sin embargo, nuevos elementos que permitan incluir una mayor consideración respecto a las prioridades del contenido, los criterios de aceptación de los usuarios finales, las preferencias del valor de negocio, así como la identificación y la asociación de los requisitos con los elementos del panorama general.

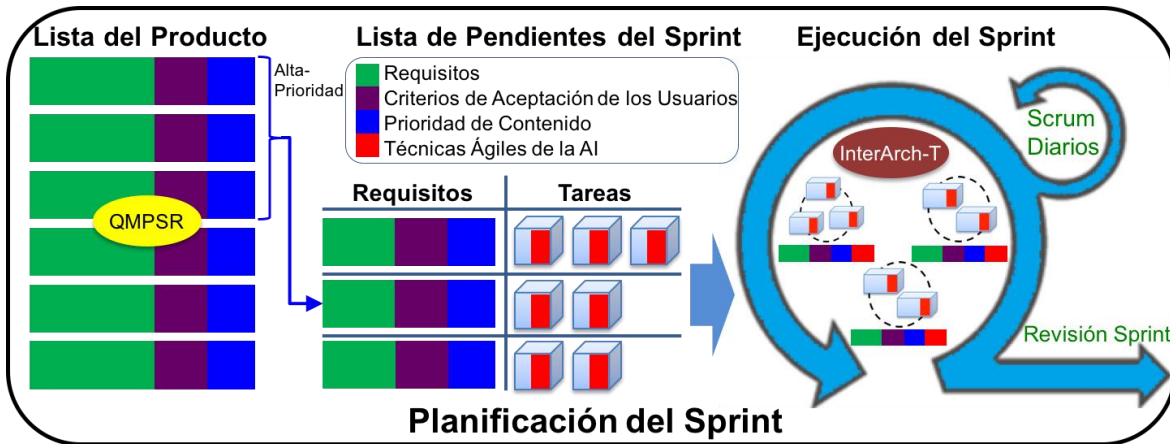
### **3.1.2. Planificación del Sprint Dirigida por la Arquitectura de la Información**

Esta segunda directriz tiene tres objetivos principales. El primer objetivo es promover, durante la Reunión de Planificación del Sprint, la implementación de los requisitos a través de las prioridades de la usabilidad, de la AI y del valor de negocio, de una manera ágil y centrada en el usuario. Con el fin de lograr el objetivo planteado, en la sección 3.3, se describe la propuesta de un método de priorización para conducir el proceso de priorización de los requisitos de Scrum-UIA a través de la prioridad de los usuarios finales, del valor de negocio y del contenido (ver QMPSR en la Figura 11).

El segundo objetivo es promover que el desarrollo se lleve a cabo de forma incremental a través de los diferentes niveles de fidelidad de los entregables de la AI (Planos, Maquetas, Asignación de Contenido, Modelos de Contenido e Inventario). Con el fin de alcanzar este objetivo, en la sección 3.4, se presenta una técnica que identifica representaciones conceptuales de la AI, que evolucionan hacia el dominio de la solución y facilitan el desarrollo incremental en Scrum-UIA, dirigido por la AI (ver InterArch-T en la Figura 11).

Finalmente, el tercer objetivo es proporcionar un entendimiento común respecto al alcance de las demandas de los requisitos presentados por el Dueño del Producto y de las tareas que se llevarán a cabo por el Equipo de Desarrollo, a través de una definición clara de “Terminado”, tal y como es recomendado en (Felker et al., 2012).

En la Figura 11, se describe la directriz de Scrum-UIA que permite conducir la Planificación del Sprint a través de la AI.



**Figura 11:** Planificación del Sprint Dirigida por la Arquitectura de la Información.

Como se muestra en la Figura 11, los requisitos de la Lista del Producto con la prioridad más alta se seleccionan para incluirlos en la Lista de Pendientes del Sprint. De este modo, cada uno de los requisitos seleccionados (que contienen, a la vez, elementos referentes a la prioridad de los usuarios finales, del valor de negocio y del contenido) se divide en tareas de desarrollo específicas por parte del Equipo de Desarrollo.

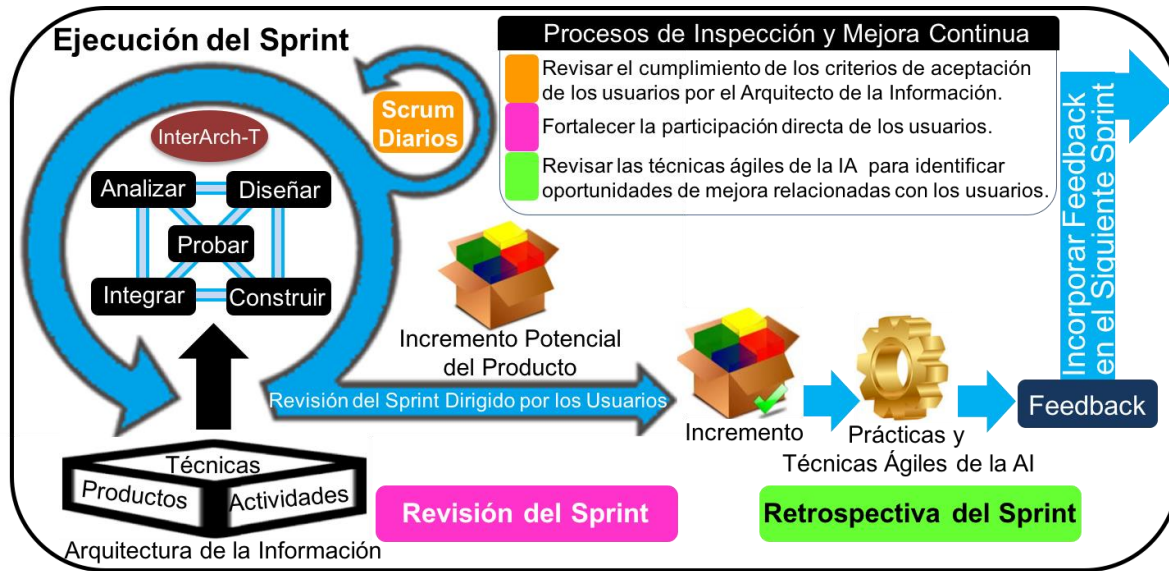
Asimismo, se proporciona un conjunto de técnicas ágiles (obtenidas desde el análisis de las nueve propuestas para la AI, ver Tabla 15) para integrar la AI, con el fin de apoyar el desarrollo de las tareas específicas durante la Ejecución del Sprint. Las técnicas ágiles para integrar la AI son auto-asignadas por el Equipo de Desarrollo, así como cada una de las tareas de desarrollo específicas.

### 3.1.3. Procesos de Inspección y Mejora Continua Dirigidos por el Usuario Final

La última directriz de Scrum-UIA tiene como objetivo fomentar el DCU y la participación de los usuarios finales en todos los procesos de inspección y mejora continua (Scrum Diario, Revisión del Sprint y Retrospectiva del Sprint).

De esta forma, durante el Scrum Diario se inspeccionan las tareas individuales, con el fin de realizar un seguimiento del cumplimiento de los criterios de aceptación de los usuarios finales, desde las primeras etapas del Sprint. Por otro lado, en la Revisión del Sprint, se evalúa el incremento potencial del producto a través de la participación directa de los usuarios finales y otros actores secundarios. Por último, pero no menos importante, durante la Retrospectiva del Sprint, se reflexiona respecto al Sprint completado para averiguar qué mejoras se podrían introducir en el siguiente Sprint.

En la Figura 12, se describe la directriz de Scrum-UIA, que permite guiar los procesos de inspección y mejora continua de Scrum-UIA a través de los usuarios finales.



**Figura 12:** Procesos de Inspección y Mejora Continua Dirigidos por el Usuario Final.

Como se muestra en la Figura 12, la Ejecución del Sprint se apoya en un conjunto de actividades, técnicas y productos para integrar la AI, de una manera ágil y centrada en el usuario. El Scrum Diario se lleva a cabo todos los días durante la Ejecución del Sprint. De este modo, se alienta y fomenta la participación del Arquitecto de la Información en el Scrum Diario para revisar las tareas del Equipo de Desarrollo, con el fin de preservar el panorama general e identificar el cumplimiento de los criterios de aceptación del usuario final.

Respecto a la Revisión del Sprint, ésta se lleva a cabo al final de la Ejecución del Sprint, con el fin de inspeccionar el incremento y gestionar la Lista del Producto (crear, definir, estimar y priorizar los requisitos), si es necesario. Se propone fortalecer la participación de los usuarios finales a través de una Revisión del Sprint dirigida por ellos, lo que podría ser también beneficioso de acuerdo con (Cajander et al., 2013). De esta manera, el Equipo de Desarrollo demuestra el trabajo finalizado a través de la participación directa del usuario final interactuando con el incremento. Por otra parte, se sugiere reflexionar sobre los resultados de estas ceremonias inmediatamente después de las evaluaciones (Felker et al., 2012) y aplicar diferentes métodos para realizar con éxito cada una de las evaluaciones centradas en el usuario (Lárusdóttir et al., 2012).

Por último, la Retrospectiva del Sprint se realiza después de la Revisión del Sprint, con el objetivo de revisar los aspectos relativos a las prácticas y técnicas ágiles de la AI, utilizadas por el Equipo de Desarrollo durante la Ejecución del Sprint. El Arquitecto de la Información utiliza este evento como una oportunidad para revisar el desempeño de las prácticas y de las técnicas utilizadas en la integración de la AI, con el fin de identificar oportunidades de mejora, con respecto a la capacidad de respuesta a las exigencias de

agilidad, y gestionar los temas relacionados con los usuarios finales. El objetivo de este evento es mejorar el proceso de Scrum-UIA, incorporando las ideas en los próximos Sprints.

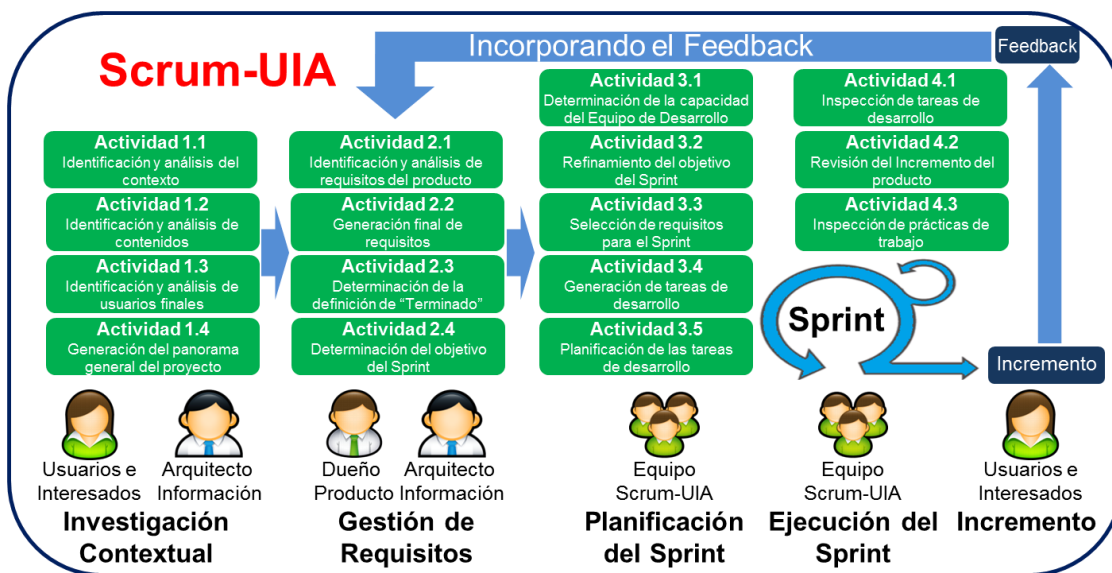
### 3.2. Proceso Scrum-UIA

Una vez presentados los lineamientos generales de Scrum-UIA, en esta sección se definen y describen las distintas actividades necesarias para abordar la construcción de aplicaciones interactivas, en base a las prioridades de la AI, dentro de un modelo de proceso ágil de desarrollo y centrado en el usuario. Estas prioridades de la AI se refieren a la necesidad de considerar la perspectiva de los usuarios finales durante todo el proceso de desarrollo de software, así como a la necesidad de conducir el proceso de priorización de los requisitos a través de las prioridades de la usabilidad, de la AI y del valor de negocio.

Los detalles completos de la propuesta de estas actividades se presentan a continuación. De este modo, en la sección 3.2.1, se presenta la descripción del marco de integración de Scrum-UIA. En la sección 3.2.2, se describe el modelo de proceso de Scrum-UIA. En la sección 3.2.3, se detalla la prescripción de todas las actividades propuestas de Scrum-UIA. Finalmente, en la sección 3.2.4, se presentan los productos generados en las diferentes actividades y tareas de Scrum-UIA.

#### 3.2.1. Descripción del Marco de Integración de Scrum-UIA

En esta sección, se detallan las actividades de la metodología Scrum-UIA, presentadas en la Figura 13. Como se puede observar, las actividades están organizadas en cuatro grupos principales: Investigación Contextual, Gestión de Requisitos, Planificación del Sprint, e Inspección y Mejora Continua.



**Figura 13:** Actividades de la metodología Scrum-UIA.

Los grupos de actividades de Investigación Contextual, Gestión de Requisitos y Planificación del Sprint se corresponden directamente con las diferentes fases propuestas en la metodología Scrum-UIA. Por otro lado, en el grupo de actividad de Inspección y Mejora Continua se organizan las actividades relacionadas con las diferentes evaluaciones, llevadas a cabo en Scrum-UIA para fomentar el DCU y la participación de los usuarios finales.

Por cada actividad de Scrum-UIA, se han especificado diferentes elementos, con el fin de que la metodología se pueda utilizar correctamente para el desarrollo de aplicaciones interactivas, dirigidas por los requisitos de los usuarios. A continuación, se presentan los elementos que se especificarán para cada una de las actividades propuestas:

- Las tareas adecuadas y las técnicas necesarias para llevar a cabo.
- La definición de los resultados o productos esperados para cada tarea.
- Los responsables de llevar a cabo las tareas.
- La relación, flujo, nivel de interdependencia entre las actividades y tareas.

La estructura de las actividades de Scrum-UIA es la siguiente:

#### **Grupo de Actividades de la Investigación Contextual**

- **Actividad 1.1. Identificación y análisis del contexto:** En esta actividad se analizan las características tanto internas como externas de la organización, relacionadas con el ámbito económico, tecnológico, legal y socio-cultural.
- **Actividad 1.2. Identificación y análisis de contenidos:** En esta actividad se identifican y analizan los contenidos relacionados con el sistema, así como las prioridades del proyecto relacionadas con el contenido.
- **Actividad 1.3. Identificación y análisis de usuarios finales:** En esta actividad se identifican y analizan todos los usuarios finales del sistema, con el fin de determinar sus características y necesidades principales.
- **Actividad 1.4. Generación del panorama general del proyecto:** En esta actividad se genera una visión global e integradora del producto, con respecto a las prioridades de los contenidos (identificadas en la Actividad 1.2), del valor de negocio (identificado en la Actividad 1.1) y de los usuarios finales (identificados en la Actividad 1.3).

#### **Grupo de Actividades de la Gestión de Requisitos**

- **Actividad 2.1. Identificación y análisis de requisitos del proyecto:** En esta actividad se identifican y analizan los requisitos funcionales del sistema, identificando los usuarios finales y el valor de negocio que aportan (identificados en la Actividad 1.4).

- **Actividad 2.2. Generación final de requisitos:** En esta actividad se describen, analizan, estiman y priorizan los requisitos del proyecto (identificados en la Actividad 2.1), con el fin de crear la Lista del Producto.
- **Actividad 2.3. Determinación de la definición de “Terminado”:** En esta actividad se identifica y determina una definición de “Terminado” para los requisitos del siguiente Sprint (identificados en la Actividad 2.2).
- **Actividad 2.4. Determinación del objetivo del Sprint:** En esta actividad se identifica y define un objetivo inicial para el siguiente Sprint, con el fin de iniciar la Planificación del Sprint con un objetivo relacionado con las prioridades de los usuarios finales.

#### **Grupo de Actividades de la Planificación del Sprint**

- **Actividad 3.1. Determinación de la capacidad del Equipo de Desarrollo:** En esta actividad se identifica y determina la velocidad y la disponibilidad de los recursos humanos que tendrá el Equipo de Desarrollo para el siguiente Sprint, con el fin de establecer su duración.
- **Actividad 3.2. Refinamiento del objetivo del Sprint:** En esta actividad se determina el objetivo, identificado inicialmente en la Actividad 2.4, que finalmente se utilizará para guiar la Ejecución del Sprint.
- **Actividad 3.3. Selección de requisitos para el Sprint:** En esta actividad se seleccionan los requisitos, generados en la Actividad 2.2, que el Equipo de Desarrollo realizará en la Ejecución del Sprint, y que forman parte de la Lista de Pendientes del Sprint.
- **Actividad 3.4. Generación de tareas de desarrollo:** En esta actividad se identifican y definen las tareas específicas de desarrollo, que se requieren para implementar los requisitos (generados en la Actividad 2.2 y seleccionados en la Actividad 3.3) del siguiente Sprint.
- **Actividad 3.5. Planificación de las tareas de desarrollo:** En esta actividad se organiza la ejecución de las tareas de desarrollo específicas (identificadas en la Actividad 3.4) para promover el desarrollo incremental a través de los niveles de fidelidad de los entregables de la AI.

#### **Grupo de Actividades de Inspección y Mejora Continua**

- **Actividad 4.1. Inspección de tareas de desarrollo:** En esta actividad se revisa el estado de las tareas de desarrollo (identificadas en la actividad 3.4 y planificadas en la Actividad 3.5) y los problemas que tienen los miembros del Equipo de Desarrollo para llevarlas a cabo, con el fin de realizar un seguimiento del cumplimiento de los criterios de aceptación de los usuarios finales.



- **Actividad 4.2. Revisión del Incremento del producto:** En esta actividad se revisa e inspecciona el incremento del producto, generado por el Equipo de Desarrollo durante la Ejecución del Sprint con la participación de los usuarios finales.
- **Actividad 4.3 Inspección de prácticas de trabajo:** En esta actividad se reflexiona respecto al Sprint completado, con el fin de indagar mejoras para el siguiente Sprint e identificar oportunidades de mejora relacionadas con los usuarios finales.

### 3.2.2. Descripción del Modelo de Proceso de Scrum-UIA

Una vez descrito el marco de integración de Scrum-UIA, en esta sección se describirán todos los detalles del modelo de proceso de Scrum-UIA.

#### 3.2.2.1. Iteración

En la metodología Scrum-UIA, la ejecución de las actividades se lleva a cabo mediante un flujo iterativo e incremental. El desarrollo del grupo de actividades, así como las actividades y las tareas que se incluyen, se ejecutan de manera secuencial o paralela, dependiendo de cada caso. Se ha elaborado la Figura 14 para ilustrar el diagrama de actividades de la metodología Scrum-UIA, con el fin de mostrar, de manera gráfica, las interdependencias entre las actividades y tareas. Por ejemplo, las actividades 1.1, 1.2 y 1.3 del *Grupo de Actividades - Investigación Contextual* pueden ejecutarse de manera paralela, debido a que sus tareas iniciales centran su análisis en aspectos diferentes, que no son interdependientes. Por el contrario, la actividad 1.4 tiene una interdependencia con las actividades 1.1, 1.2 y 1.3, lo que significa que la elaboración del panorama general del proyecto (Actividad 1.4) requiere que previamente se hayan ejecutado las actividades 1.1, 1.2 y 1.3. Es decir, el diseño correcto del panorama general requiere como entrada la identificación y análisis del contexto (Actividad 1.1), la identificación y análisis del contenido (Actividad 1.2), y la identificación y análisis de los usuarios finales (Actividad 1.3). Lo anterior se puede observar de forma gráfica a través del nodo anterior a la actividad 1.4, que actúa como sincronización y derivación. Por un lado, el nodo en el sentido de sincronización permite representar un punto intermedio de revisión de diferentes tareas, como en el caso de la sincronización de las tareas 1.1.2, 1.3.2 y 1.3.2. Por otro lado, el nodo de derivación permite representar ejecuciones de tareas de modo paralelo, como en el caso de la derivación de las tareas 1.4.1 y 1.4.2 de la actividad 1.4.

En el *Grupo de Actividades – Gestión de Requisitos* la ejecución de la actividad 2.1 es interdependiente de la actividad 1.4 del *Grupo de Actividades - Investigación Contextual*. Esto ocurre debido a que el panorama general del proyecto, generado por el *Grupo de Actividades - Investigación Contextual*, permite guiar la actividad 2.1 sobre la identificación y análisis de los requisitos del producto. En el *Grupo de Actividades – Gestión de Requisitos*, se observa que después de la tarea 2.2.2 existe un nodo de derivación, debido a que las tareas sobre la estimación del tamaño de los requisitos (tarea

2.2.3) y la priorización de los requisitos (tarea 2.2.4) no son interdependientes, y se pueden ejecutar de forma paralela. Solamente se requiere finalizar la tarea 2.2.2 que genera el conjunto de requisitos del producto. Asimismo, se puede observar que después de la actividad 2.2 existe un nodo de derivación hacia las actividades 2.3 y 2.4, debido a que la determinación tanto de la definición de “Terminado” como del objetivo del Sprint no son tareas interdependientes, y éstas pueden llevarse a cabo una vez generado el conjunto final de requisitos (actividad 2.2).

En el *Grupo de Actividades – Planificación del Sprint*, la ejecución de las actividades 3.1 y 3.2 son interdependientes de las actividades 2.3 y 2.4 del *Grupo de Actividades – Gestión de Requisitos*. Esto se puede observar a través del nodo que existe después de las actividades 2.3 y 2.4, lo que significa que el inicio de la Planificación del Sprint está supeditado a la determinación de un objetivo inicial para el Sprint y un acuerdo sobre la definición de “Terminado” para las tareas de desarrollo. Similarmente, la selección de los requisitos para el Sprint (Actividad 3.3) es interdependiente de las actividades 3.1 y 3.2 (ver nodo de sincronización después de las tareas 3.1.3 y 3.2.1), lo que significa que la elaboración de la Lista de Pendientes del Sprint (Actividad 3.3) requiere que previamente se hayan definido la capacidad del Equipo de Desarrollo (Actividad 3.1) y el objetivo del Sprint (Actividad 3.2). Finalmente, la actividad 3.5 es interdependiente de las tareas 3.4.2, 3.4.3 y 3.4.4 de la actividad 3.4 (ver el nodo de sincronización que está después de las tareas), lo que indica que la planificación de las tareas de desarrollo (Actividad 3.5) demanda que éstas se encuentren estimadas, asignadas a sus responsables y asociadas a las técnicas ágiles de la AI (correspondientes a las tareas 3.4.2, 3.4.3 y 3.4.4, respectivamente).

Por último, en el *Grupo de Actividades – Inspección y Mejora Continua*, la actividad 4.1 es una actividad de control de las tareas de desarrollo y su ejecución se espera durante los días del Sprint.

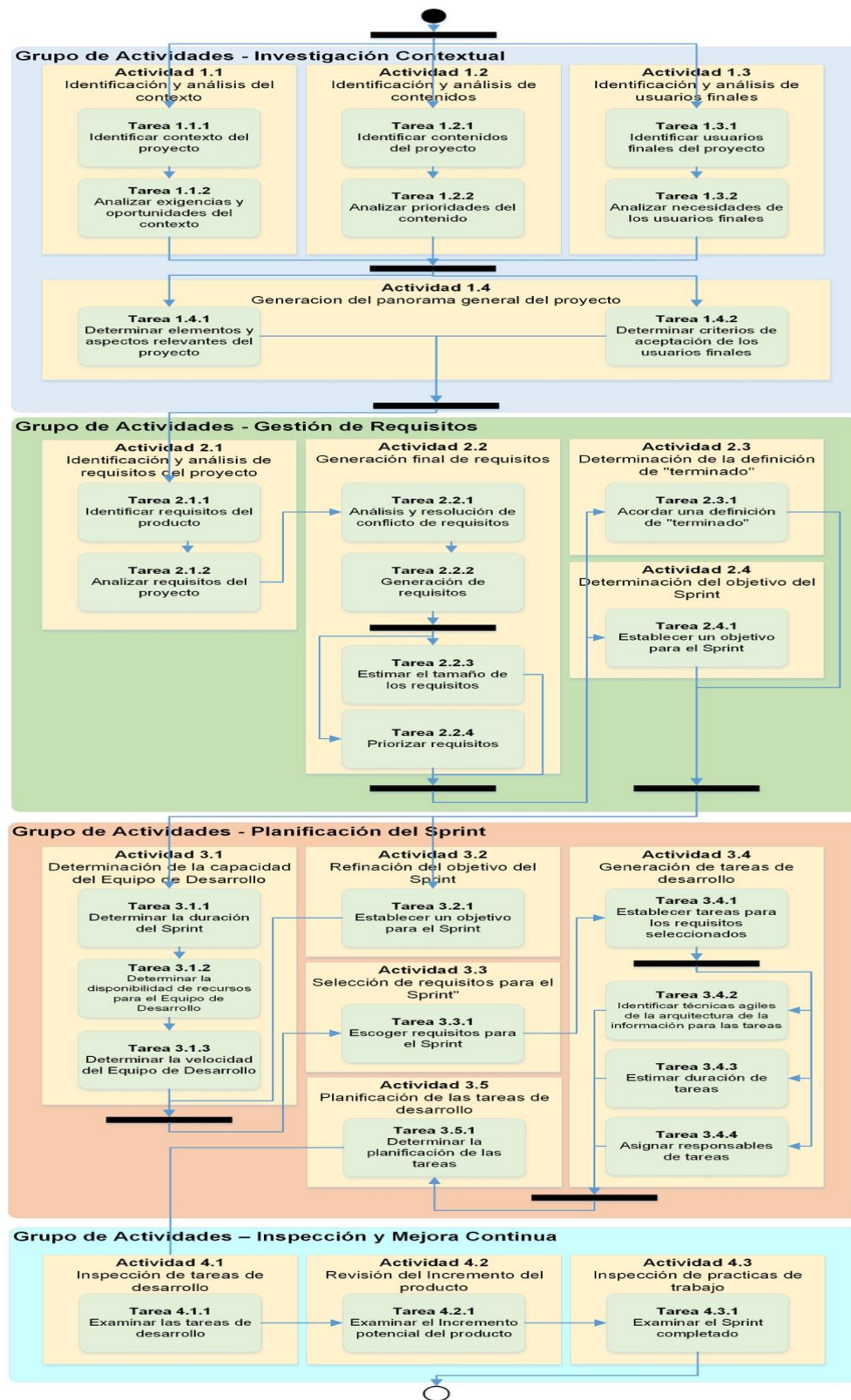


Figura 14: Diagrama de actividad con la interdependencia entre las actividades y tareas propuestas.

### **3.2.2.2. Flujos de Información**

La ejecución de cada una de las tareas propuestas en las actividades del modelo genera diferentes flujos de información, que a la vez se utilizan por otras tareas dentro del modelo. Los flujos de información se pueden clasificar en información interna y externa. Los flujos de información interna corresponden a la información que es generada por otras tareas del modelo, por ejemplo, resultados de análisis de una tarea determinada, entre otros. Por otro lado, los flujos de información externa consisten en información que procede de entidades externas al proceso de desarrollo, por ejemplo, conocimientos provenientes de profesionales que están involucrados en el proceso, datos históricos del producto, entre otros. Para una ejecución correcta de las tareas, es necesario que tanto la información interna como la externa sea gestionada de manera eficiente.

La Figura 15 se ha elaborado para ilustrar el flujo de información interna de todas las tareas propuestas. Como se puede observar en la Figura 15, existe un gran flujo de información interna entre las tareas, lo que demuestra el alto nivel de interdependencia existente. Dicho flujo de información permite explicar su dependencia. Por otro lado, la ausencia de flujo de información permite demostrar que las tareas se pueden llevar a cabo de forma paralela, sin que existan interdependencias entre ellas, como, por ejemplo, en el caso de las tareas 1.1.1, 1.2.1 y 1.3.1. El caso contrario corresponde a la tarea 1.4.1 que recibe los flujos de información de las tareas 1.1.2, 1.2.2 y 1.3.2. Esto permite demostrar que la tarea 1.4.1 es interdependiente de la ejecución de las tareas 1.1.2, 1.2.2 y 1.3.2.

Los flujos de información interna del modelo se plasman a través de los diferentes productos que se generan y se utilizan entre las diversas tareas de las actividades. Por ejemplo, la tarea 1.1.1 de la actividad 1.1 genera como producto de salida un informe sobre las características principales del contexto del producto (P1). A la vez, este producto se utiliza como información de entrada en la tarea 1.1.2 de la actividad 1.1, y en las tareas 1.4.1 y 1.4.2 de la actividad 1.4, generando un flujo de información entre dichas tareas. Esto puede visualizarse en ilustración de la Figura 15, con las flechas que nacen desde la tarea 1.1.1 y se enlazan hacia las tareas 1.1.2, 1.4.1 y 1.4.2.

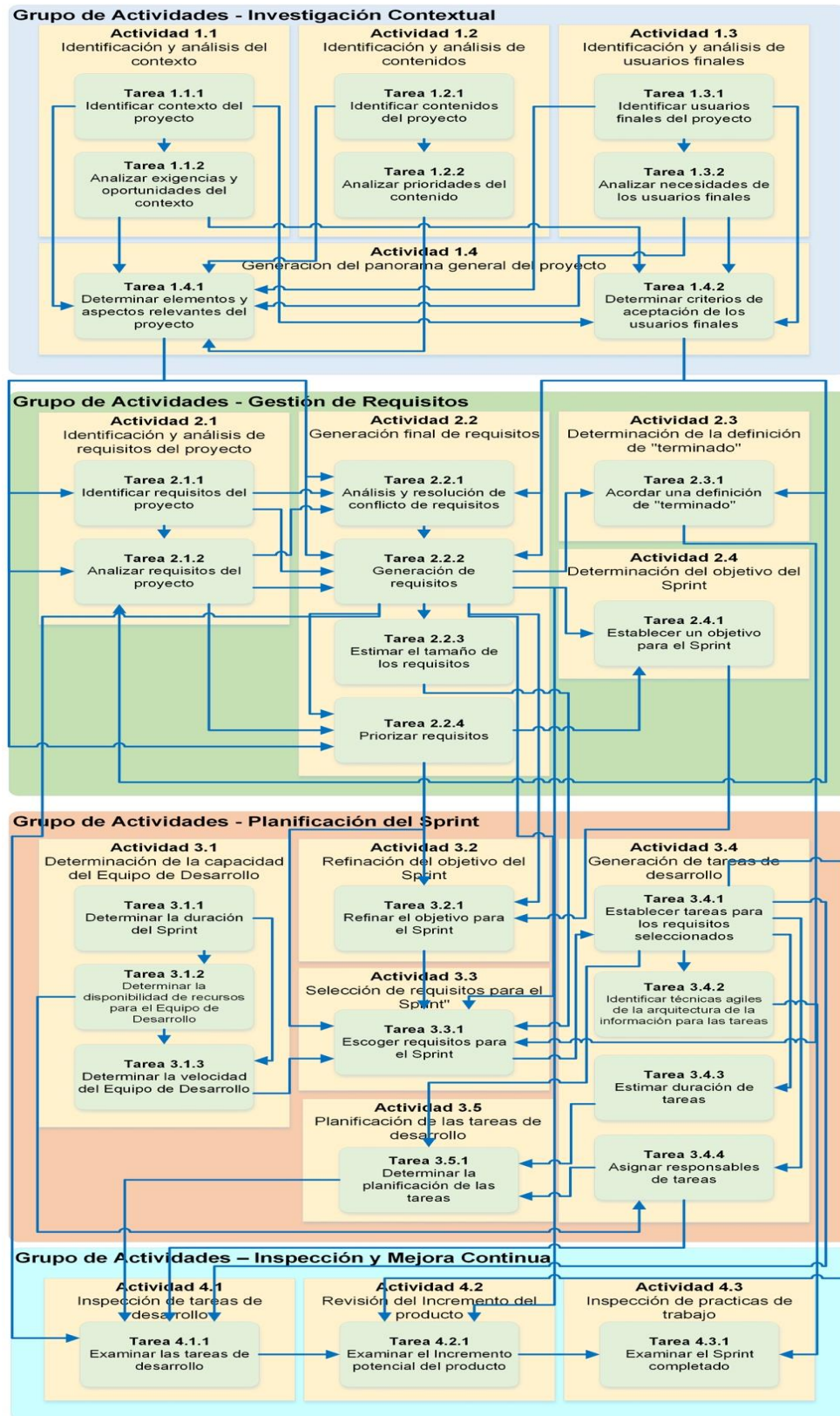


Figura 15: Flujo de Información Interna de cada una de las tareas de las actividades.

### 3.2.2.3. Notación

En esta sección se presentan las notaciones utilizadas para describir las tareas, las actividades y los productos del modelo propuesto, que se detallan a continuación:

#### a) Estructura

La estructura de la propuesta consiste en tres niveles: *Grupo de Actividades*, *Actividades* y *Tareas*. Su descripción se presenta a continuación:

- **Grupo de Actividades:** Este nivel corresponde al grupo de actividades de la metodología Scrum-UIA, que integra un conjunto de actividades con un propósito común. Por ejemplo, el grupo de actividades de la gestión de requisitos agrupa un conjunto de actividades que tienen como objetivo común llevar a cabo la correcta gestión de los elementos de la Lista del Producto.
  - **Actividad:** Este nivel corresponde a la descripción y al detalle de las actividades, propuestas por cada *Grupo de Actividades*.
    - **Tarea:** Este nivel corresponde a la descripción de las tareas que se requieren, con el fin de llevar a cabo las *Actividades* propuestas.

#### b) Descripción

A continuación, se describen los detalles de los elementos propuestos anteriormente:

- **Grupo de Actividades:** Para cada *Grupo de Actividades* se detallan tanto su objetivo como las *Actividades* que los componen.
- **Actividades:** Para cada *Actividad* se detallan su objetivo y las diferentes *Tareas* que se proponen.
- **Tareas:** Para cada *Tarea* se describen los siguientes elementos:
  - **Descripción:** Corresponde a la descripción de la tarea, donde se indican los productos de entrada que son requeridos, la información externa necesaria y los productos de salida que se deben generar.
  - **Técnicas y Métodos:** Corresponde a las técnicas y métodos recomendados para llevar a cabo las tareas propuestas. Las técnicas y métodos sugeridos para las tareas son descritos en la sección 3.2.2.4.
  - **Participantes:** Corresponde a las personas que están involucradas en la ejecución y gestión de la tarea. Entre los participantes se distingue quién es el responsable de la tarea con la letra *R* (R). Los diferentes participantes de las tareas son descritos en la sección 3.2.2.5.
- **Información de Entrada:** Corresponde a la información que necesita la tarea para su correcta ejecución. La información de entrada puede provenir de dos tipos de fuentes:
  - a) Interna del proceso de desarrollo, que proviene desde los productos creados por otras tareas.

- b) Externa al proceso de desarrollo, que no es generada por ninguna de las tareas del proceso y que proviene, por ejemplo, del conocimiento de las personas que participan en el proyecto, la documentación establecida del proyecto, entre otras.

Para cada información de entrada de las tareas se especificará lo siguiente:

- **ID:** Corresponde a un identificador único de la información de entrada. Este identificador distingue entre la información de entrada externa e interna, asignando la nomenclatura *Ex* para la información de entrada externa, y la nomenclatura *Px* para la información de entrada interna. El *x* representa un número correlativo y único, asignado para cada uno de los tipos de información utilizados en el proceso de desarrollo (es decir, *E* y *P* para la información externa e interna, respectivamente).
- **Descripción:** Este elemento puede corresponder a dos tipos:
  - En el caso de que sea información interna (*Px*), contendrá la descripción del nombre del producto.
  - En el caso de que sea información externa (*Ex*), este elemento describirá, en términos generales, las características de la información externa requerida.
- **Fuente:** Se identifica la tarea que generó el producto requerido (por ejemplo: Tarea 1.1.1), o si la fuente de la información es externa (por ejemplo: Externa).
- **Productos de Salida:** Corresponde al conjunto de productos, generados como resultado de la ejecución de la tarea. Para cada ítem de esta lista se detallará lo siguiente:
  - **ID:** Corresponde al código identificador del producto con el formato *Px*, donde *x* representa un número correlativo y único, asignado para cada uno de los productos.
  - **Producto:** Corresponde al nombre del producto, de acuerdo a la nomenclatura definida.
  - **Destino:** Corresponde a la tarea de destino, que utilizará el producto descrito.

#### 3.2.2.4. Técnicas y Métodos Recomendados

Se incluyen recomendaciones de técnicas y métodos para el desarrollo correcto de las tareas, propuestas para cada una de las actividades. Las principales técnicas y métodos recomendados son:

**QMPSR:** Es el método de priorización de requisitos que se propone en esta tesis doctoral, con el fin de guiar el proceso de priorización de manera ágil a través de las prioridades de la usabilidad, de la AI y del valor de negocio. Este método de priorización permite capturar

formalmente las prioridades de cada proyecto, a través de elementos cualitativos que definen las preferencias de los requisitos. Estas prioridades se identifican y se generan de manera dinámica y colectiva entre los diferentes participantes del proceso de priorización. En la sección 3.3, se describen los detalles completos de este método de priorización.

**InterArch-T:** Es la técnica que se propone en esta tesis doctoral para fomentar el desarrollo incremental en Scrum-UIA mediante la AI. Esta técnica permite representar el conocimiento proveniente de las definiciones conceptuales del Arquitecto de la Información, así como su posterior transformación, utilizando reglas explícitas, en información de análisis y diseño para Analistas e Ingenieros del Software. En la sección 3.4, se presentan los detalles de esta técnica.

**Entrevistas no estructuradas:** En una entrevista, los encuestados son interrogados en un diálogo personal entre el entrevistador y el entrevistado. En el caso de las entrevistas no estructuradas, éstas no tienen preguntas predefinidas y las respuestas son abiertas (DiCicco-Bloom y Crabtree, 2006).

**Cuestionario:** Es un conjunto de preguntas escritas o impresas, entregadas al encuestado. Por lo tanto, no hay necesidad de que el analista esté presente durante su realización. Los cuestionarios pueden ser cerrados (proporcionando una gama de posibles respuestas a cada pregunta) o abiertos (sin rango de respuestas proporcionadas) (Oppenheim, 2000).

**Investigación de antecedentes:** La investigación de antecedentes es una actividad para revisar los antecedentes previos que están relacionados con el proyecto. Se utilizan comúnmente para revisar los datos históricos de la organización y del contexto del proyecto (Morville y Rosenfeld, 2006).

**Benchmarking:** Es una comparación de aspectos específicos de una organización con los líderes o competidores más importantes del mercado. Esta técnica permite identificar, evaluar y comparar, de manera sistemática, las características de un proyecto (Vorhies y Morgan, 2005).

**Reunión:** Esta técnica permite recoger información a través del acto, por el que un conjunto de personas se reúne con un objetivo común (Montgomery y Strick, 1994).

**Evaluación heurística:** La evaluación heurística consiste en identificar inconvenientes de usabilidad en el diseño de una interfaz de usuario (González et. al, 2006). Esta técnica implica revisar la interfaz por evaluadores expertos en usabilidad, de acuerdo a los principios de usabilidad reconocidos (heurísticas) (Nielsen, 1994).

**Clasificación de tarjetas:** La clasificación de tarjetas es un método, que permite identificar la estructura latente en una lista sin ordenar los elementos de ideas o contenido. Este método consiste en indicar cada elemento en una pequeña tarjeta para, a continuación, solicitar a los participantes ordenarlas y/o clasificarlas en grupos. Los resultados



individuales se pueden combinar y, si es necesario, se analizan estadísticamente (Gaffney, 2000).

**Historias de usuario:** Las historias de usuarios son un instrumento para representar los requisitos de un proyecto software. Cada historia de usuario se describe en términos sencillos y simples, desde el punto de vista del usuario o del cliente que requiere la funcionalidad. Normalmente, en la descripción de las historias de usuarios, se identifica al usuario o al cliente, al objetivo deseado y al valor de negocio que aporta a la organización (Cohn, 2004).

**Planificación póker:** La planificación póker es una técnica ágil de planificación, basada en el consenso. Esta técnica permite estimar el esfuerzo de la implementación de los requisitos de un proyecto a través de un conjunto de tarjetas enumeradas (normalmente utilizando la secuencia de Fibonacci e incluyendo un cero, es decir, 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89). De esta forma, cada participante dispone de un juego de tarjetas, y en la estimación de cada requisito todos vuelven boca arriba las tarjetas, en lugar de hablarlas en voz alta (Cohn, 2005).

**Puntos de historia:** Los puntos de historia permiten dimensionar y relacionar la complejidad de los requisitos con respecto a otros. Los puntos de historia son una unidad de medida relativa para expresar el tamaño global de un requisito. Por ejemplo, un requisito estimado en cuatro puntos de historia indica que es el doble de grande, complejo o riesgoso que un requisito estimado con dos puntos de historia. Asimismo, un requisito estimado en seis puntos de historia es la mitad de grande, complejo o riesgoso que un requisito de doce puntos de historia (Coelho y Basu, 2012).

### 3.2.2.5. Participantes

Las tareas propuestas para cada una de las actividades contienen la identificación de las diferentes personas que participan en la ejecución y gestión de las tareas. A continuación, se describen los principales participantes que se proponen en la definición de cada una de las tareas de las actividades:

- **El Dueño del Producto (*Product Owner*):** Es el responsable de gestionar (crear, definir, estimar y priorizar) los elementos de la Lista del Producto a través del trabajo en colaboración y coordinación con el Arquitecto de la Información, con el fin de maximizar el valor del producto y del trabajo del Equipo de Desarrollo. No obstante, cabe destacar que el Dueño del Producto es el responsable final de gestionar la Lista del Producto.
- **El Equipo de Desarrollo (*Development Team*):** El Equipo de Desarrollo está compuesto por profesionales con conocimiento multidisciplinar, orientados a entregar un incremento de producto “Terminado”.

- **El Scrum-UIA Master:** El Scrum-UIA Master tiene la responsabilidad de que la teoría, las prácticas y las reglas de la metodología Scrum-UIA sean entendidas y adoptadas de forma correcta, tanto por el equipo Scrum-UIA como por la organización.
- **El Arquitecto de la Información:** El Arquitecto de la Información es el responsable de dirigir la Investigación Contextual, apoyar la gestión de la Lista del Producto, promover el desarrollo incremental a través de los niveles de fidelidad de los entregables de la AI, asegurar que el diseño sea centrado en el usuario y fomentar la involucración de los usuarios finales.
- **Usuarios Finales:** Son las personas que van a interactuar directamente con el programa computacional (software).
- **Stakeholders:** Son todas aquellas personas, grupos o entidades que son afectadas o pueden ser afectadas con el producto que se desea desarrollar.

### 3.2.3. Prescripción de las Actividades Propuestas

A continuación, se presenta la prescripción de las actividades, de acuerdo a la estructura propuesta anteriormente.

#### 3.2.3.1. Grupo de Actividades de la Investigación Contextual

Este grupo de actividades tiene como objetivo obtener conocimientos sobre las prioridades de los usuarios finales, del valor de negocio y del contenido, de acuerdo al contexto del proyecto. Asimismo, se proporcionan las bases para establecer un panorama general del proyecto, con el fin de que la perspectiva de los usuarios finales se pueda discutir, describir y considerar a través del modelo de proceso de Scrum-UIA. Las actividades para este grupo corresponden a:

##### Actividad 1.1. Identificación y análisis del contexto

**Descripción:** El objetivo de esta actividad es identificar y analizar las exigencias y oportunidades del contexto. Por lo tanto, en esta actividad se deben identificar los objetivos a corto y largo plazo, así como los presupuestos, los programas, la infraestructura tecnológica, los recursos humanos y la cultura corporativa del proyecto.

Esta actividad se inicia con la identificación del contexto (Tarea 1.1.1), siendo el Arquitecto de la Información el responsable para llevarla a cabo. La Tarea 1.1.1 sólo utiliza información externa (E1), correspondiente a la información recabada por el Arquitecto de la Información a través de entrevistas no estructuradas, presentaciones y cuestionarios a los stakeholders, así como a través de investigaciones sobre los antecedentes del proyecto. El principal resultado de la Tarea 1.1.1 consiste en un Informe sobre las características principales del contexto (P1), dando cuenta sobre el entorno en que se desarrollará el proyecto.

Posteriormente, en la Tarea 1.1.2, el Arquitecto de la Información utiliza las técnicas de benchmarking, reuniones y entrevistas no estructuradas para analizar las amenazas y oportunidades del contexto, con el fin de identificar los aspectos más importantes del entorno donde se llevará a cabo el proyecto. El Arquitecto de la Información es el responsable de la Tarea 1.1.2, y su principal fuente de información corresponde al Informe sobre las características principales del contexto (P1), provisto por la Tarea 1.1.1. El principal resultado de la Tarea 1.1.2 consiste en un Informe sobre las amenazas y oportunidades del contexto (P2).

La actividad 1.1, sobre la identificación y análisis del contexto, tiene dos resultados principales: el primero corresponde a un Informe sobre las características principales del contexto (P1), y el segundo es un Informe sobre las amenazas y oportunidades del contexto (P2). Ambos sirven de información de entrada para analizar y configurar los aspectos y elementos del panorama general del proyecto, relacionados con el negocio.

A continuación, se detallan las distintas tareas necesarias para la realización de la actividad:

#### **Tarea 1.1.1. Identificar contexto del proyecto**

**Descripción:** Identificar las características principales del contexto, donde acontece el proyecto.

**Técnicas:** Investigación de Antecedentes, Presentaciones, Reuniones de Investigación, Entrevistas No Estructuradas.

**Participantes:** Arquitecto de la Información (R<sup>13</sup>), Stakeholders.

ID	Información de entrada	Fuente
E1	Las características del contexto, donde ocurre el proyecto (los objetivos, presupuestos, programas, infraestructura tecnológica, recursos humanos y la cultura corporativa)	Externa

ID	Productos de salida	Destino
P1	Informe sobre las características principales del contexto	Tarea 1.4.1, Tarea 1.4.2

#### **Tarea 1.1.2. Analizar exigencias y oportunidades del contexto**

**Descripción:** Analizar las oportunidades y las exigencias del contexto, con el fin de identificar los aspectos más relevantes del entorno del proyecto.

---

<sup>13</sup> Esta letra (R) identifica al responsable de la tarea, tal y como se describió en la sección 3.2.2.3.

**Técnicas:** Benchmarking, Reuniones, Entrevistas No Estructuradas.

**Participantes:** Arquitecto de la Información (R), Stakeholders.

ID	Información de Entrada	Fuente
P1	Informe sobre las características principales del contexto	Tarea 1.1.1, Tarea 1.4.1 Tarea 1.4.2

ID	Productos de Salida	Destino
P2	Informe sobre las amenazas y oportunidades del contexto	Tarea 1.4.1, Tarea 1.4.2

### Actividad 1.2. Identificación y análisis de contenidos

**Descripción:** En esta actividad se debe identificar y analizar todo el contenido relacionado con el proyecto, que puede tener distintas formas (imágenes, archivos de videos, documentos, gráficos, archivos de audio, logotipos, archivos de mensajes de correo electrónico, entre otros). El objetivo de esta actividad es determinar las prioridades relacionadas con los contenidos, que se deben considerar en el desarrollo del Sistema de Información (SI).

Esta actividad se inicia con la identificación de los contenidos asociados al proyecto (Tarea 1.2.1), donde el Arquitecto de la Información es el responsable de llevarla a cabo. La Tarea 1.2.1 sólo utiliza información externa (E2), que consiste en los objetos de contenido, en sus diferentes formatos asociados al proyecto. El Arquitecto de la Información recopila la información externa a través de entrevistas no estructuradas, reuniones al personal de gestión de contenidos, e investigaciones de antecedentes, relacionados con el contenido del proyecto. Asimismo, el Arquitecto de la Información utiliza la herramienta InterArch (esta herramienta sistematiza la técnica InterArch-T, que se describe en detalle en la sección 3.4) para generar un modelo de contenido inicial, con el fin de representar visualmente las relaciones y dependencias más importantes de los contenidos. Los dos principales resultados de la Tarea 1.2.1 consisten en un Informe sobre la identificación de los contenidos del proyecto (P3) y un Modelo de contenido general del proyecto (P4).

Posteriormente, en la Tarea 1.2.2, se analiza el contenido relacionado con el proyecto, con el fin de entender su alcance, origen y prioridades. El Arquitecto de la Información es el responsable de la Tarea 1.2.2 y utiliza las técnicas de reuniones, clasificación de tarjetas y evaluación heurística para realizar los diferentes análisis relacionados con el contenido. Las principales fuentes de información de la Tarea 1.2.2 son el Informe sobre la identificación de los contenidos del proyecto (P3) y un Modelo de contenido general del proyecto (P4),

ambos proporcionados por la tarea 1.2.1. El principal resultado de la Tarea 1.2.2 es un Informe sobre el análisis de las prioridades del contenido (P5).

La actividad 1.2, sobre la identificación y análisis de contenidos, tiene tres resultados principales: un Informe sobre la identificación de los contenidos del proyecto (P3), un Modelo de contenido general del proyecto (P4) y un Informe sobre el análisis de las prioridades del contenido (P5).

A continuación, se detallan las distintas tareas necesarias para la realización de la actividad:

#### **Tarea 1.2.1. Identificar contenidos del proyecto**

**Descripción:** Identificar los documentos y objetos de contenido asociados al proyecto.

**Técnicas:** Investigación de antecedentes, Entrevistas No Estructuradas, Reuniones, InterArch-T.

**Participantes:** Arquitecto de la Información (R), Stakeholders.

ID	Información de Entrada	Fuente
E2	Los documentos y objetos de contenido del proyecto, en sus diferentes formatos	Externa

ID	Productos de Salida	Destino
P3	Informe sobre la identificación de los contenidos del proyecto	Tarea 1.2.2, Tarea 1.4.1
P4	Modelo de contenido general del proyecto	Tarea 1.2.2, Tarea 1.4.1, Tarea 3.4.1

#### **Tarea 1.2.2. Analizar prioridades del contenido**

**Descripción:** Analizar los objetos de contenido del proyecto, con el fin de identificar las prioridades y elementos más importantes del contenido.

**Técnicas:** Evaluación Heurística, Clasificación de Tarjetas, Reuniones, Entrevistas No Estructuradas.

**Participantes:** Arquitecto de la Información (R), Stakeholders.

ID	Información de Entrada	Fuente
P3	Informe sobre la identificación de los contenidos del proyecto	Tarea 1.2.1

P4	Modelo de contenido general del proyecto	Tarea 1.2.1
----	--	-------------

ID	Productos de Salida	Destino
P5	Informe sobre el análisis de las prioridades del contenido	Tarea 1.4.1

### Actividad 1.3. Identificación y análisis de usuarios finales

**Descripción:** Esta actividad tiene como objetivo identificar y analizar los usuarios finales del SI.

En la Tarea 1.3.1, se identifican los usuarios finales del SI y sus características y necesidades relacionadas con el proyecto, siendo el Arquitecto de la Información el responsable de llevar dicha tarea a cabo. La Tarea 1.3.1 sólo utiliza información externa (E3), que consiste en los usuarios finales del proyecto y sus características principales. El Arquitecto de la Información utiliza diferentes prácticas para identificar y obtener los rasgos más importantes de los usuarios finales, tales como revisar los registros de los datos de soporte de los clientes, analizar el registro de búsqueda de los usuarios finales, revisar los resultados sobre estadísticas de usos de productos similares, entre otros. El principal resultado de la Tarea 1.3.1 es un Informe preliminar de los usuarios finales del proyecto y sus características (P6).

Posteriormente, en la Tarea 1.3.2, el Arquitecto de la Información lleva a cabo reuniones y entrevistas no estructuradas, para analizar las necesidades más importantes de los usuarios finales, identificados en la tarea anterior (Tarea 1.3.1). El Arquitecto de la Información es el responsable de conducir el análisis de la Tarea 1.3.2, y la fuente principal de información consiste en el Informe preliminar de los usuarios finales del proyecto y sus características (P6), provisto por la Tarea 1.3.1. La Tarea 1.3.2 tiene como principal resultado un Informe final sobre las necesidades de los usuarios finales (P7).

La Actividad 1.3 sobre la identificación y análisis de usuarios finales tiene dos resultados principales: el primero corresponde a un Informe preliminar de los usuarios finales del proyecto y sus características (P6), y el segundo es un Informe final sobre las necesidades de los usuarios finales (P7).

A continuación, se detallan las distintas tareas, necesarias para la realización de la actividad:

#### Tarea 1.3.1. Identificar usuarios finales del proyecto

**Descripción:** Identificar a todos los usuarios finales y sus características y necesidades relacionadas con el proyecto.

**Técnicas:** Entrevistas No Estructuradas, Cuestionarios.

**Participantes:** Arquitecto de la Información (R), Stakeholders, Usuarios finales.

ID	Información de Entrada	Fuente
E3	Los usuarios finales del producto y sus características	Externa

ID	Productos de Salida	Destino
P6	Informe preliminar de los usuarios finales del proyecto y sus características	Tarea 1.4.1, Tarea 1.4.2

#### **Tarea 1.3.2. Analizar necesidades de los usuarios finales**

**Descripción:** Analizar las necesidades de los usuarios finales (identificados en la Tarea 1.3.1), relacionadas con el proyecto.

**Técnicas:** Reuniones, Entrevistas No Estructuradas.

**Participantes:** Arquitecto de la Información (R), Usuarios Finales.

ID	Información de Entrada	Fuente
P6	Informe preliminar de los usuarios finales del proyecto y sus características	Tarea 1.3.1

ID	Productos de Salida	Destino
P7	Informe final sobre las necesidades de los usuarios finales	Tarea 1.4.1, 1.4.2

#### **Actividad 1.4. Generación del panorama general del proyecto**

**Descripción:** En esta actividad se debe determinar el panorama general del proyecto, que permita obtener una visión global e integradora del SI, con respecto a las prioridades de la usabilidad, de los contenidos, del valor de negocio y de los usuarios finales.

En la Tarea 1.4.1, se determinan los elementos y los aspectos relevantes del proyecto que permiten discernir el nivel de prioridad de los requisitos (estos posteriormente se utilizan para conducir el proceso de priorización de Scrum-UIA, mediante el método de priorización QMPSR, descrito en detalle en la sección 3.3). El Arquitecto de la Información es el responsable de esta tarea, y las fuentes principales de información provienen del Informe sobre las características principales del contexto (P1) de la Tarea 1.1.1, del Informe sobre las amenazas y oportunidades del contexto (P2) de la Tarea 1.1.2, del Informe sobre la identificación de los contenidos del proyecto (P3) de la Tarea 1.2.1, del Modelo de

contenido general del proyecto (P4) de la Tarea 1.2.1, del Informe sobre el análisis de las prioridades del contenido (P5) de la Tarea 1.2.2, del Informe preliminar de los usuarios finales del proyecto y sus características (P6) de la Tarea 1.3.1, y del Informe final sobre las necesidades de los usuarios finales (P7) de la Tarea 1.3.2. El Arquitecto de la Información (responsable) y el Dueño del Producto trabajan en conjunto a través de reuniones, para definir los elementos y los aspectos relevantes del proyecto. El principal resultado de la Tarea 1.4.1 es un Informe final sobre los elementos y los aspectos relevantes del proyecto (P8).

Finalmente, en la Tarea 1.4.2, se determinan los criterios de aceptación de los usuarios finales. El Arquitecto de la Información es el responsable de esta tarea, y las fuentes principales de información provienen del Informe sobre las características principales del contexto (P1) de la Tarea 1.1.1, del Informe sobre las amenazas y oportunidades del contexto (P2) de la Tarea 1.1.2, del Informe preliminar de los usuarios finales del proyecto y sus características (P6) de la Tarea 1.3.1, y del Informe final sobre las necesidades de los usuarios finales (P7) de la Tarea 1.3.2. El Arquitecto de la Información realiza reuniones con los usuarios finales para determinar sus criterios de aceptación (P9), lo que corresponde al resultado principal de esta tarea.

La Actividad 1.4, sobre la generación del panorama general del proyecto, tiene dos resultados principales: el primero consiste en un Informe final sobre los elementos y los aspectos relevantes del proyecto (P8), y el segundo en un Catálogo de criterios de aceptación de los usuarios finales (P9). Ambos resultados permiten configurar el panorama general del proyecto y dirigir todo el proceso de acuerdo a las prioridades de la usabilidad, de los contenidos, del valor de negocio y de los usuarios finales.

A continuación, se detallan las distintas tareas, necesarias para la realización de la actividad:

#### **Tarea 1.4.1. Determinar los elementos y los aspectos relevantes del proyecto**

**Descripción:** Establecer los aspectos relevantes del proyecto, y cuáles son los elementos que lo componen. El objetivo de esta tarea es determinar las prioridades del proyecto (elementos y aspectos relevantes del proyecto) que permiten conducir el proceso de priorización de los requisitos.

**Técnicas:** Reuniones.

**Participantes:** Arquitecto de la Información (R), Dueño del Producto.

ID	Información de Entrada	Fuente
P1	Informe sobre las características principales del contexto	Tarea 1.1.1



P2	Informe sobre las amenazas y oportunidades del contexto	Tarea 1.1.2
P3	Informe sobre la identificación de los contenidos del proyecto	Tarea 1.2.1
P4	Modelo de contenido general del proyecto	Tarea 1.2.1
P5	Informe sobre el análisis de las prioridades del contenido	Tarea 1.2.2
P6	Informe preliminar de los usuarios finales del proyecto y sus características	Tarea 1.3.1
P7	Informe final sobre las necesidades de los usuarios finales	Tarea 1.3.2

ID	Productos de Salida	Destino
P8	Informe final sobre los elementos y los aspectos relevantes del proyecto	Tarea 2.1.1, Tarea 2.1.2, Tarea 2.2.1, Tarea 2.2.2, Tarea 2.2.4

#### **Tarea 1.4.2. Determinar criterios de aceptación de los usuarios finales**

**Descripción:** Determinar las características de usabilidad que debe tener el producto para cumplir con la conformidad y exigencias de los usuarios finales.

**Técnicas:** Reuniones.

**Participantes:** Arquitecto de la Información (R), Usuarios Finales.

ID	Información de Entrada	Fuente
P1	Informe sobre las características principales del contexto	Tarea 1.1.1
P2	Informe sobre las amenazas y oportunidades del contexto	Tarea 1.1.2
P6	Informe preliminar de los usuarios finales del proyecto y sus características	Tarea 1.3.1
P7	Informe final sobre las necesidades de los usuarios finales	Tarea 1.3.2

ID	Productos de Salida	Destino
P9	Catálogo de criterios de aceptación de los usuarios finales	Tarea 2.1.2, Tarea 2.2.1, Tarea 2.2.2, Tarea 2.3.1

### 3.2.3.2. Grupo de Actividades de la Gestión de Requisitos

Este grupo de actividades tiene como objetivo identificar, analizar, priorizar y estimar los requisitos del proyecto, y está dirigido por el conocimiento, obtenido en el Grupo de Actividades de la Investigación Contextual, de tal forma que los requisitos del proyecto reflejen el valor de negocio, las prioridades de los usuarios finales y del contenido. Las actividades para este grupo corresponden a:

#### Actividad 2.1. Identificación y análisis de requisitos del proyecto

**Descripción:** En esta actividad se deben identificar y analizar los requisitos del proyecto.

Esta actividad se inicia con la Tarea 2.1.1, que tiene como objetivo identificar los requisitos del proyecto, donde el Dueño del Producto representa y provee las necesidades e intereses del negocio, siendo además el responsable de llevarla a cabo. La Tarea 2.1.1 utiliza información del Informe final sobre los elementos y los aspectos relevantes del proyecto (P8) de la Tarea 1.4.1, y Catálogo de criterios de aceptación de los usuarios finales (P9) de la Tarea 1.4.2. Ambos informes permiten guiar en la obtención de la tercera información de entrada, correspondiente a los requisitos funcionales del proyecto (E5). El Dueño del Producto, en colaboración con el Arquitecto de la Información, lleva a cabo reuniones, entrevistas no estructuradas y cuestionarios, para identificar los requisitos de los usuarios finales y de los stakeholders. Más específicamente, el Arquitecto de la Información participa en esta tarea, con el fin de proveer la visión del panorama general del proyecto, obtenida desde el Grupo de Actividades de la Investigación Contextual. El principal resultado de la Tarea 2.1.1 consiste en un Informe preliminar de requisitos del proyecto (P10).

Posteriormente, en la Tarea 2.1.2, el Dueño del Producto, en colaboración con el Arquitecto de la Información, realiza reuniones para analizar los requisitos del proyecto identificados, con el fin clasificarlos y determinar el valor de negocio que aportan. El Dueño del Producto es responsable de la Tarea 2.1.2, y las fuentes de información de esta tarea corresponden al Informe final sobre los elementos y los aspectos relevantes del proyecto (P8) de la Tarea 1.4.1, el Catálogo de criterios de aceptación de los usuarios finales (P9) de la Tarea 1.4.2 y el Informe preliminar de requisitos del proyecto (P10) de la Tarea 2.1.1. El principal resultado de la Tarea 1.4.2 corresponde a un Catálogo de requisitos analizados del proyecto (P11).

La actividad 2.1, sobre la identificación y análisis de requisitos del proyecto, tiene dos resultados principales: el primero es el Informe preliminar de requisitos del proyecto (P10), y el segundo es el Catálogo de requisitos analizados del proyecto (P11). Ambos permiten formar la información de base para configurar la generación final de los requisitos del producto.

A continuación, se detallan las distintas tareas, necesarias para la realización de la actividad:

#### **Tarea 2.1.1. Identificar requisitos del proyecto**

**Descripción:** Identificar y caracterizar individualmente las necesidades e intereses de los usuarios finales y de los stakeholders relacionados con el proyecto.

**Técnicas:** Reuniones, Entrevistas No Estructuradas, Cuestionarios.

**Participantes:** Dueño del Producto (R), Arquitecto de la Información, Stakeholders, Usuarios Finales.

ID	Información de Entrada	Fuente
E5	Los requisitos funcionales del proyecto	Externa
P8	Informe final sobre los elementos y los aspectos relevantes del proyecto	Tarea 1.4.1
P9	Catálogo de criterios de aceptación de los usuarios finales	Tarea 1.4.2

ID	Productos de Salida	Destino
P10	Informe preliminar de requisitos del proyecto	Tarea 2.1.2, Tarea 2.2.1, Tarea 2.2.2

#### **Tarea 2.1.2. Analizar requisitos del proyecto**

**Descripción:** Analizar los requisitos del proyecto, identificados en la Tarea 2.1.1, con el fin de clasificarlos e identificar el valor de negocio que proporcionan.

**Técnicas:** Reuniones.

**Participantes:** Dueño del Producto (R), Arquitecto de la Información.

ID	Información de Entrada	Fuente
P8	Informe final sobre los elementos y los aspectos relevantes del proyecto	Tarea 1.4.1
P9	Catálogo de criterios de aceptación de los usuarios finales	Tarea 1.4.2
P10	Informe preliminar de requisitos del proyecto	Tarea 2.1.1

ID	Productos de Salida	Destino
P11	Catálogo de requisitos analizados del proyecto	Tarea 2.2.1, Tarea 2.2.2, Tarea 2.2.4

### Actividad 2.2. Generación final de requisitos

**Descripción:** En esta actividad se deben describir, analizar, estimar y priorizar los requisitos del proyecto que se han identificado anteriormente en la actividad 2.1. El objetivo de esta actividad es generar una Lista del Producto, donde sus elementos (historias de usuarios) consideren las prioridades de los contenidos, del valor de negocio, de los criterios de aceptación de los usuarios finales, así como se identifiquen y se asocien con los elementos del panorama general del proyecto.

Esta actividad se inicia con la Tarea 2.2.1, que tiene como objetivo analizar y resolver los conflictos de los requisitos. En esta tarea, el Dueño del Producto y el Arquitecto de la Información trabajan en conjunto para identificar, analizar y resolver los conflictos y dependencias entre los requisitos del proyecto. La Tarea 2.2.1 utiliza diferentes fuentes de información, tales como el Informe final sobre los elementos y los aspectos relevantes del proyecto (P8) de la Tarea 1.4.1, el Catálogo de criterios de aceptación de los usuarios finales (P9) de la Tarea 1.4.2, el Informe preliminar de requisitos del proyecto (P10) de la Tarea 2.1.1 y el Catálogo de requisitos analizados del proyecto (P11) de la Tarea 2.1.2. El principal resultado de la Tarea 2.2.1 es el Catálogo final revisado de requisitos del proyecto (P12).

Después, en la Tarea 2.2.2, los requisitos del proyecto se configuran utilizando la Lista del Producto, donde sus elementos se describen en formato de historias de usuario. En esta actividad, el Dueño del Producto es el responsable final de la generación correcta de la Lista del Producto. En la Tarea 2.2.2, el Arquitecto de la Información también participa para identificar y asociar los requisitos con los elementos del panorama general del proyecto, con el fin de que los requisitos tengan asociados los criterios de aceptación de los usuarios finales y los aspectos principales relacionados con el contenido. En la descripción

de los requisitos del proyecto, se utiliza la técnica de historias de usuarios, con el fin de identificar a los usuarios finales, a sus objetivos y al valor de negocio que aporta cada requisito. La Tarea 2.2.2 utiliza como información de entrada los siguientes productos: el Informe final sobre los elementos y los aspectos relevantes del proyecto (P8) de la Tarea 1.4.1, el Catálogo de criterios de aceptación de los usuarios finales (P9) de la Tarea 1.4.2, el Informe preliminar de requisitos del proyecto (P10) de la Tarea 2.1.1, el Catálogo de requisitos analizados del proyecto (P11) de la Tarea 2.1.2 y el Catálogo final revisado de requisitos del proyecto (P12) de la Tarea 2.2.1. El principal resultado de la Tarea 2.2.2 corresponde a una Lista del Producto con los elementos descritos en formato de Historia de Usuario (P13).

Una vez finalizada la tarea anterior, en la Tarea 2.2.3 se estima el tamaño de los requisitos de la Lista del Producto, donde el Equipo de Desarrollo es el responsable de esta tarea, pero todos los miembros de Scrum-UIA participan, gracias a la utilización de la técnica de planificación póker. De este modo, los valores de las estimaciones son indicados, preferentemente, en puntos de historias o días ideales (días-persona). La Tarea 2.2.3 utiliza como fuente principal de información la Lista del Producto con los elementos descritos en formato de Historia de Usuario (P13), proporcionada por la Tarea 2.2.2. El principal resultado de la Tarea 2.2.3 corresponde a un Informe de la estimación de los elementos de la Lista del Producto (P14).

Finalmente, en la Tarea 2.2.4, se priorizan los requisitos del producto, en un trabajo en coordinación y colaboración entre el Dueño del Producto y el Arquitecto de la Información. En la realización de la Tarea 2.2.4, se utiliza un método de priorización (en la sección 3.3, se presenta su descripción completa), que permite conducir el proceso mediante el uso de elementos cualitativos, que están relacionados con las prioridades del proyecto (identificadas en la Tarea 1.4.1). Por un lado, el Arquitecto de la Información prioriza los requisitos de la Lista del Producto, proporcionando la visión de la AI y de la usabilidad, obtenida desde el Grupo de Actividades de la Investigación Contextual. El Arquitecto de la Información utiliza como fuentes de información el Informe final sobre los elementos y los aspectos relevantes del proyecto (P8) de la Tarea 1.4.1, y el Catálogo de criterios de aceptación de los usuarios finales (P9) de la Tarea 1.4.2. Específicamente, el Arquitecto de la Información tiene como objetivo ponderar con una mayor prioridad a los requisitos de la Lista del Producto, que evolucionan hacia el dominio de la solución y facilitan el desarrollo incremental, dirigido por la AI (en la sección 3.4, se describe cómo conducir el desarrollo incremental a través de los entregables de la AI).

Por otro lado, el Dueño del Producto prioriza los requisitos de la Lista del Producto, proporcionando la visión de las necesidades y de las prioridades del negocio (obtenidas desde el Catálogo de requisitos analizados del proyecto (P11) de la Tarea 2.1.2). Tanto el Dueño del Producto como el Arquitecto de la Información trabajan en la priorización de los

requisitos sobre la Lista del Producto con los elementos descritos en formato de Historia de Usuario (P13), proporcionada por la Tarea 2.2.2 y el Informe final sobre los elementos y los aspectos relevantes del proyecto (P8) de la Tarea 1.4.1. La Tarea 2.2.4 tiene como resultado principal un Informe de las prioridades de los elementos de la Lista del Producto (P15). Este producto de salida permite identificar los elementos de los aspectos relevantes, que están asociados con los requisitos y las prioridades, identificadas tanto por el Arquitecto de la Información como por el Dueño del Producto para cada uno de los requisitos.

La actividad 2.2 sobre la generación final de requisitos conlleva tres resultados principales: la Lista del Producto con los elementos descritos en formato de Historia de Usuario (P13) de la Tarea 2.2.2, un Informe de la estimación de los elementos de la Lista del Producto (P14) de la Tarea 2.2.3 y un Informe de las prioridades de los elementos de la Lista del Producto (P15) de la Tarea 2.2.4. Todos estos resultados generan, en conjunto, una Lista del Producto, descrita en Historia de Usuario, estimada y priorizada.

A continuación, se detallan las distintas tareas, necesarias para la realización de la actividad:

**Tarea 2.2.1. Análisis y resolución de conflicto de requisitos**

**Descripción:** Analizar la dependencia y conflicto entre los distintos requisitos del proyecto.

**Técnicas:** Reunión.

**Participantes:** Dueño del Producto (R), Arquitecto de la Información.

ID	Información de Entrada	Fuente
P8	Informe final sobre los elementos y los aspectos relevantes del proyecto	Tarea 1.4.1
P9	Catálogo de criterios de aceptación de los usuarios finales	Tarea 1.4.2
P10	Informe preliminar de requisitos del proyecto	Tarea 2.1.1
P11	Catálogo de requisitos analizados del proyecto	Tarea 2.1.2

ID	Productos de Salida	Destino
P12	Catálogo final revisado de requisitos del proyecto	Tarea 2.2.2, Tarea 2.2.4, Tarea 3.2.1, Tarea 3.3.1

### Tarea 2.2.2. Generación de requisitos

**Descripción:** Registrar los requisitos del proyecto en la Lista del Producto.

**Técnicas:** Historias de Usuario.

**Participantes:** Dueño del Producto (R), Arquitecto de la Información.

ID	Información de Entrada	Fuente
P8	Informe final sobre los elementos y los aspectos relevantes del proyecto	Tarea 1.4.1
P9	Catálogo de criterios de aceptación de los usuarios finales	Tarea 1.4.2
P10	Informe preliminar de requisitos del proyecto	Tarea 2.1.1
P11	Catálogo de requisitos analizados del proyecto	Tarea 2.1.2
P12	Catálogo final revisado de requisitos del proyecto	Tarea 2.2.1

ID	Productos de Salida	Destino
P13	Lista del Producto con los elementos descritos en formato de Historia de Usuario	Tarea 2.3.1, Tarea 2.4.1, Tarea 3.2.1, Tarea 3.3.1, Tarea 4.1.1, Tarea 4.2.1

### Tarea 2.2.3. Estimar el tamaño de los requisitos

**Descripción:** Estimar el tamaño de los requisitos del proyecto.

**Técnicas:** Planificación Póker, Puntos de Historias.

**Participantes:** Equipo de Desarrollo (R), Dueño del Producto, Arquitecto de la Información, Scrum-UIA Master.

ID	Información de Entrada	Fuente
P13	Lista del Producto con los elementos descritos en formato de Historia de Usuario	Tarea 2.2.2

ID	Productos de Salida	Destino
P14	Informe de la estimación de los elementos de la Lista del Producto	Tarea 3.2.1, Tarea 3.3.1

#### Tarea 2.2.4. Priorizar requisitos

**Descripción:** Priorizar los requisitos del proyecto en base a las prioridades de los usuarios finales, del contenido y del valor de negocio.

**Técnicas:** QMPSR.

**Participantes:** Dueño del Producto (R), Arquitecto de la Información.

ID	Información de Entrada	Fuente
P8	Informe final sobre los elementos y los aspectos relevantes del proyecto	Tarea 1.4.1
P9	Catálogo de criterios de aceptación de los usuarios finales	Tarea 1.4.2
P11	Catálogo de requisitos analizados del proyecto	Tarea 2.1.2
P13	Lista del Producto con los elementos descritos en formato de Historia de Usuario	Tarea 2.2.2

ID	Productos de Salida	Destino
P15	Informe de las prioridades de los elementos de la Lista del Producto	Tarea 2.4.1, Tarea 3.2.1, Tarea 3.3.1

#### Actividad 2.3. Determinación de la definición de “Terminado”

**Descripción:** En esta actividad se debe identificar y determinar una definición de “Terminado” para los requisitos del siguiente Sprint, con el objetivo de obtener un entendimiento común sobre el alcance y las demandas de los requisitos, presentados por el Dueño del Producto (en colaboración con el Arquitecto de la Información, y definidos en la Actividad 2.2), y de las tareas que se llevarán a cabo por el Equipo de Desarrollo.

Esta actividad sólo tiene la Tarea 2.3.1, que tiene como objetivo identificar y acordar, entre el equipo Scrum-UIA completo, una definición de “Terminado” para el siguiente Sprint, siendo el Dueño del Producto el responsable de esta tarea. Para lograr el objetivo de la Tarea 2.3.1, los participantes utilizan la técnica de la reunión, para identificar y acordar la definición de “Terminado”. Dicha tarea utiliza como fuente de información la Lista del



Producto con los elementos descritos en formato de Historia de Usuario (P13) de la Tarea 2.2.2, y la priorización de los elementos de la Lista del Producto (P15), proporcionada por la Tarea 2.2.4. El resultado principal de la Tarea 2.3.1 consiste en un Informe sobre la definición de “Terminado” para el Sprint (P16).

A continuación, se detalla la tarea necesaria para la realización de la actividad:

**Tarea 2.3.1. Acordar una definición de “Terminado”**

**Descripción:** Determinar una definición de “Terminado” para el siguiente Sprint.

**Técnicas:** Reuniones.

**Participantes:** Dueño del Producto (R), Arquitecto de la Información, Equipo de Desarrollo, Scrum-UIA Master.

ID	Información de Entrada	Fuente
P9	Catálogo de criterios de aceptación de los usuarios finales	Tarea 1.4.2
P13	Lista del Producto con los elementos descritos en formato de Historia de Usuario	Tarea 2.2.2

ID	Productos de Salida	Destino
P16	Informe sobre la definición de “Terminado” para el Sprint	Tarea 3.2.1

**Actividad 2.4. Determinación del objetivo del Sprint**

**Descripción:** En esta actividad se debe identificar y determinar un objetivo inicial para el siguiente Sprint, con la finalidad de comenzar la Planificación del Sprint con un objetivo relacionado con las prioridades de los usuarios finales, propuesto por el Dueño del Producto y el Arquitecto de la Información.

Esta actividad sólo tiene la Tarea 2.4.1, que propone un objetivo inicial para el siguiente Sprint y que se puede modificar durante la Planificación del Sprint. En la Tarea 2.4.1, el Dueño del Producto es el responsable, y utiliza como fuente de información la Lista del Producto con los elementos descritos en formato de Historia de Usuario (P13) de la Tarea 2.2.2, y la estimación de los elementos de la Lista del Producto (P14) de la Tarea 2.2.4. El resultado principal de la Tarea 2.4.1 consiste en un Informe sobre el objetivo inicial para el Sprint (P17).

A continuación, se detalla la tarea necesaria para la realización de la actividad:

#### **Tarea 2.4.1. Establecer un objetivo para el Sprint**

**Descripción:** Determinar un objetivo para el siguiente Sprint.

**Técnicas:** Reuniones.

**Participantes:** Dueño del Producto (R), Arquitecto de la Información.

ID	Información de Entrada	Fuente
P13	Lista del Producto con los elementos descritos en formato de Historia de Usuario	Tarea 2.2.2
P15	Informe de las prioridades de los elementos de la Lista del Producto	Tarea 2.2.4

ID	Productos de Salida	Destino
P17	Informe sobre el objetivo inicial para el Sprint	Tarea 3.2.1

#### **3.2.3.3. Grupo de Actividades de la Planificación del Sprint**

Este grupo de actividades tiene como objetivo identificar, estimar y planificar las tareas de desarrollo que son necesarias para implementar los requisitos seleccionados para el Sprint (obtenidos desde el Grupo de Actividades de la Gestión de Requisitos). Las actividades para este grupo corresponden a:

##### **Actividad 3.1. Determinación de la capacidad del Equipo de Desarrollo**

**Descripción:** En esta actividad se debe definir la duración del siguiente Sprint y determinar la capacidad del Equipo de Desarrollo de acuerdo a su disponibilidad de recursos.

Esta actividad se inicia con la Tarea 3.1.1 que consiste en determinar la duración del Sprint. Esta duración es establecida por el equipo Scrum-UIA completo durante la Reunión de Planificación del Sprint, que debería estar en el rango entre 1 y 4 semanas. En la Tarea 3.1.1, sólo se utiliza información externa, correspondiente a la duración de los Sprints anteriores (E6), y tiene como principal información de salida el Informe sobre la duración del Sprint (P18).

Después, en la Tarea 3.1.2, el Equipo de Desarrollo identifica y determina la disponibilidad de recursos humanos que tendrá para el siguiente Sprint. La Tarea 3.1.2 utiliza como fuentes de entrada el Informe sobre la duración del Sprint (P18) de la Tarea 3.1.1 y las capacidades disponibles del Equipo de Desarrollo (E7) para calcular y determinar las disponibilidades de cada miembro del Equipo de Desarrollo para el siguiente Sprint. La Tarea 3.1.2 tiene como principal resultado el Informe sobre la disponibilidad del Equipo de Desarrollo para el Sprint (P19).

Finalmente, en la Tarea 3.1.3, se determina la velocidad del Equipo de Desarrollo para el siguiente Sprint. La Tarea 3.1.3 se lleva a cabo durante la Reunión de Planificación del Sprint, y es el Equipo de Desarrollo el encargado de determinar su propia velocidad. La Tarea 3.1.3 utiliza como fuentes de entrada la velocidad histórica del Equipo de Desarrollo (E8), obtenida desde información de los Sprints anteriores, el Informe sobre la duración del Sprint (P18) de la Tarea 3.1.1, y el Informe sobre la disponibilidad del Equipo de Desarrollo para el Sprint (P19) de la Tarea 3.1.2. La Tarea 3.1.3 tiene como principal salida Informe sobre la velocidad del Equipo de Desarrollo (P20).

La actividad 3.1 sobre la determinación de la capacidad del Equipo de Desarrollo tiene tres resultados principales: el Informe sobre la duración del Sprint (P18), el Informe sobre la disponibilidad del Equipo de Desarrollo para el Sprint (P19) y el Informe sobre la velocidad del Equipo de Desarrollo (P20). Los resultados de esta actividad facilitan la determinación del compromiso que el Equipo de Desarrollo puede adquirir para el siguiente Sprint, y es la información de base para validar la consistencia de la estimación de las tareas específicas de desarrollo.

A continuación, se detallan las distintas tareas, necesarias para la realización de la actividad:

#### **Tarea 3.1.1. Determinar la duración del Sprint**

**Descripción:** Establecer la duración del Sprint que el Equipo de Desarrollo requiere para generar un incremento del producto.

**Técnicas:** Reunión de Planificación del Sprint.

**Participantes:** Dueño del Producto, Arquitecto de la Información, Equipo de Desarrollo (R), Scrum-UIA Master.

ID	Información de Entrada	Fuente
E6	La duración de los Sprints anteriores	Externa

ID	Productos de Salida	Destino
P18	Informe sobre la duración del Sprint	Tarea 3.1.2, Tarea 3.1.3

#### **Tarea 3.1.2. Determinar la disponibilidad de recursos para el Equipo de Desarrollo**

**Descripción:** Identificar y determinar la disponibilidad de recursos humanos que tendrá el Equipo de Desarrollo para el siguiente Sprint.

**Técnicas:** Reunión de Planificación del Sprint.

**Participantes:** Equipo de Desarrollo (R), Scrum-UIA Master.

ID	Información de Entrada	Fuente
E7	Las capacidades disponibles del Equipo de Desarrollo	Externa
P18	Informe sobre la duración del Sprint	Tarea 3.1.1

ID	Productos de Salida	Destino
P19	Informe sobre la disponibilidad del Equipo de Desarrollo para el Sprint	Tarea 3.1.3, Tarea 3.4.4

### **Tarea 3.1.3. Determinar la velocidad del Equipo de Desarrollo**

**Descripción:** Calcular y determinar la velocidad del Equipo de Desarrollo para el siguiente Sprint.

**Técnicas:** Reunión de Planificación del Sprint, Puntos de Historia.

**Participantes:** Equipo de Desarrollo (R), Scrum-UIA Master.

ID	Información de Entrada	Fuente
E8	La velocidad histórica del Equipo de Desarrollo	Externa
P18	Informe sobre la duración del Sprint	Tarea 3.1.1
P19	Informe sobre la disponibilidad del Equipo de Desarrollo para el Sprint	Tarea 3.1.2

ID	Productos de Salida	Destino
P20	Informe sobre la velocidad del Equipo de Desarrollo	Tarea 3.3.1

### **Actividad 3.2. Refinamiento del objetivo del Sprint**

**Descripción:** En esta actividad se debe refinar y determinar el objetivo final para el siguiente Sprint que permitirá guiar su ejecución.

Esta actividad sólo tiene la Tarea 3.2.1, que tiene como fin refinar y establecer, entre todos los miembros del equipo Scrum-UIA, el objetivo que finalmente se utilizará para el siguiente Sprint. En la Tarea 3.2.1, el Dueño del Producto es el responsable, pero, a diferencia de la Tarea 2.4.1 (Actividad 2.4), esta vez participa el equipo Scrum-UIA completo. Para lograr el objetivo de la Tarea 3.2.1, los participantes utilizan la técnica de reunión para analizar el objetivo inicialmente propuesto (P17) por el Dueño del Producto en la Tarea 2.4.1. Además, la Tarea 3.2.1 utiliza como fuentes de información la Lista del

Producto con los elementos descritos en formato de Historia de Usuario (P13) de la Tarea 2.2.2 y el Informe de la estimación de los elementos de la Lista del Producto (P14) de la Tarea 2.2.4. El resultado principal de la Tarea 3.2.1 consiste en Informe sobre el objetivo refinado del Sprint (P21).

A continuación, se detalla la tarea necesaria para la realización de la actividad:

#### **Tarea 3.2.1. Refinar el objetivo para el Sprint**

**Descripción:** Determinar un objetivo para el siguiente Sprint.

**Técnicas:** Reuniones.

**Participantes:** Dueño del Producto (R), Arquitecto de la Información, Equipo de Desarrollo, Scrum-UIA Master.

ID	Información de Entrada	Fuente
P13	Lista del Producto con los elementos descritos en formato de Historia de Usuario	Tarea 2.2.2
P15	Informe de las prioridades de los elementos de la Lista del Producto	Tarea 2.2.4
P17	Informe sobre el objetivo inicial para el Sprint	Tarea 2.4.1

ID	Productos de Salida	Destino
P21	Informe sobre el objetivo refinado del Sprint	Tarea 3.3.1

#### **Actividad 3.3. Selección de requisitos para el Sprint**

**Descripción:** En esta actividad se deben seleccionar todos los requisitos que se implementarán en el siguiente Sprint. Estos deben considerar las capacidades del Equipo de Desarrollo (determinadas en la actividad 3.1) y las prioridades de los requisitos (definidas en la actividad 2.2).

Esta actividad sólo tiene la Tarea 3.3.1, que tiene como objetivo escoger los requisitos que el Equipo de Desarrollo se comprometerá a ejecutar en el siguiente Sprint. En la Tarea 3.2.1, el Equipo de Desarrollo es el responsable de llevarla a cabo. De este modo, el Equipo de Desarrollo trabaja sobre una Lista del Producto que se encuentra descrita en formato de historias de usuarios, estimada y priorizada (P13, P14 y P15). Por lo tanto, se debe comenzar con el requisito superior de la Lista del Producto (con mayor prioridad) y posteriormente pasar al siguiente, y así sucesivamente, hasta alcanzar la capacidad del Equipo de Desarrollo (P20). El principal resultado de la Tarea 3.3.1 es una Lista de Pendientes del Sprint con los requisitos para el Sprint (P22).

A continuación, se detalla la tarea necesaria para la realización de la actividad:

#### **Tarea 3.3.1. Escoger requisitos para el Sprint**

**Descripción:** Determinar y escoger los requisitos para el siguiente Sprint, que conformarán la Lista de Pendientes del Sprint.

**Técnicas:** Reunión de Planificación del Sprint.

**Participantes:** Equipo de Desarrollo.

ID	Información de Entrada	Fuente
P13	Lista del Producto con los elementos descritos en formato de Historia de Usuario	Tarea 2.2.2
P14	Informe de la estimación de los elementos de la Lista del Producto	Tarea 2.2.3
P15	Informe de las prioridades de los elementos de la Lista del Producto	Tarea 2.2.4
P20	Informe sobre la velocidad del Equipo de Desarrollo	Tarea 3.1.3
P21	Informe sobre el objetivo refinado del Sprint	Tarea 3.2.1

ID	Productos de Salida	Destino
P22	Lista de Pendientes del Sprint con los requisitos para el Sprint	Tarea 3.4.1

#### **Actividad 3.4. Generación de tareas de desarrollo**

**Descripción:** En esta actividad, se deben identificar y definir las tareas específicas de desarrollo que se requieren para implementar los requisitos seleccionados para la Lista de Pendientes del Sprint (compuesta y definida en la actividad 3.3).

Esta actividad se inicia con la Tarea 3.4.1 que consiste en especificar las tareas de desarrollo que se requieren para alcanzar cada uno de los requisitos seleccionados para el siguiente Sprint. En la Tarea 3.4.1, el Equipo de Desarrollo es el responsable de definir cada una de las tareas. No obstante, el Arquitecto de la Información también participa asesorando al Equipo de Desarrollo para identificar las tareas relacionadas con la AI. Específicamente, el Arquitecto de la Información identifica representaciones de la AI (modelo de contenido, maquetas, planos, entre otros) que evolucionan a fases más cercanas al dominio de la solución y permiten promover desarrollo incremental en Scrum-UIA (en la sección 3.4, se describe la técnica InterArch-T que permite transformar los modelos de contenido en diagrama de clases UML, facilitando el desarrollo incremental en Scrum-UIA

a través de los niveles de fidelidad de los entregables de la AI). Las fuentes de información de la Tarea 3.4.1 son un Modelo de contenido general del proyecto de la Tarea 1.2.1, una Lista de Pendientes del Sprint con los requisitos para el Sprint (P22) de la Tarea 3.3.1, y el Informe sobre la definición de “Terminado” para el Sprint (P16) de la Tarea 2.3.1. La Tarea 3.4.1 tiene como salida principal el Informe de las tareas de desarrollo de los requisitos del Sprint (P23) y un Diagrama de Clases UML (P24). Cabe destacar que la elaboración del Modelo de contenido general del proyecto (P4) mediante InterArch (ver Tarea 1.2.1) permite generar, de manera automática, el Diagrama de Clases UML (P24). Este diagrama se puede utilizar por los miembros del Equipo de Desarrollo, para apoyar la generación de tareas de desarrollo, y durante la Ejecución del Sprint.

Una vez definidas las tareas de desarrollo, en la Tarea 3.4.2, se identifican y asignan técnicas y herramientas ágiles de la AI, para las tareas de desarrollo que lo requieran. El Arquitecto de la Información es el responsable de identificar qué tareas de desarrollo requieren técnicas y herramientas ágiles de la AI. No obstante, el Equipo de Desarrollo debe decidir su inclusión dentro de las tareas de desarrollo establecidas. La Tarea 3.4.2 utiliza como fuente externa un conjunto de técnicas ágiles para integrar la AI (E9), que es gestionada por el Arquitecto de la Información. Además, en la Tarea 3.4.2, también se utilizan como fuente de información las tareas de desarrollo para cada uno de los requisitos seleccionados para el siguiente Sprint (P23) de la Tarea 3.4.1. La Tarea 3.4.2 tiene como principal salida el Informe de las técnicas ágiles de la Arquitectura de la Información relacionadas con las tareas de desarrollo (P25).

Después, en la Tarea 3.4.3, se estima y calcula la duración de las tareas de desarrollo definidas para el Sprint. El Equipo de Desarrollo es el responsable de la Tarea 3.4.3, no obstante, el Arquitecto de la Información también participa asesorando al Equipo de Desarrollo en la estimación de la duración de las tareas relacionadas con la AI. La Tarea 3.4.3 sólo utiliza como fuente de entrada el Informe de las tareas de desarrollo de los requisitos del Sprint (P23) de la Tarea 3.4.1. El principal resultado de la Tarea 3.4.3 corresponde al Informe sobre la estimación de las tareas de desarrollo (P26).

Finalmente, en la Tarea 3.4.4, se asignan los responsables de cada una de las tareas de desarrollo descritas para alcanzar los requisitos. En esta tarea, el Equipo de Desarrollo es el responsable de llevarla a cabo, y se utilizan como fuentes de información el Informe sobre la disponibilidad del Equipo de Desarrollo para el Sprint (P19), proporcionada por la Tarea 3.1.2, y el Informe de las tareas de desarrollo de los requisitos del Sprint (P23) de la Tarea 3.4.1. La Tarea 3.4.4 tiene como principal resultado el Informe de los responsables de las tareas de desarrollo (P27).

La actividad 3.4, sobre la generación de las tareas de desarrollo, tiene cinco resultados principales: el Informe de las tareas de desarrollo de los requisitos del Sprint (P23), el Diagrama de Clases UML (P24), el Informe de las técnicas ágiles de la Arquitectura de la

Información relacionadas con las tareas de desarrollo (P25), el Informe sobre la estimación de las tareas de desarrollo (P26) y el Informe de los responsables de las tareas de desarrollo (P27). Todos los informes forman en conjunto la Lista de Pendientes del Sprint, donde sus elementos se encuentran estimados, asignados a responsables y asociados a técnicas ágiles de la AI.

A continuación, se detallan las distintas tareas, necesarias para la realización de la actividad:

#### **Tarea 3.4.1. Establecer tareas para los requisitos seleccionados**

**Descripción:** Determinar tareas específicas de desarrollo para cada uno de los requisitos seleccionados para el siguiente Sprint.

**Técnicas:** Reunión de Planificación del Sprint, InterArch-T.

**Participantes:** Equipo de Desarrollo (R), Arquitecto de la Información.

ID	Información de Entrada	Fuente
P4	Modelo de contenido general del proyecto	Tarea 1.2.1
P22	Lista de Pendientes del Sprint con los requisitos para el Sprint	Tarea 3.3.1
P16	Informe sobre la definición de “Terminado” para el Sprint	Tarea 2.3.1

ID	Productos de Salida	Destino
P23	Informe de las tareas de desarrollo de los requisitos del Sprint	Tarea 3.4.2, Tarea 3.4.3, Tarea 3.4.4, Tarea 3.5.1, Tarea 4.1.1, Tarea 4.2.1
P24	Diagrama de Clases UML	Ejecución del Sprint

#### **Tarea 3.4.2. Identificar técnicas ágiles de la Arquitectura de la Información para las tareas**

**Descripción:** Identificar y asignar, cada vez que se requiera, técnicas ágiles de la AI para las tareas de desarrollo específicas.

**Técnicas:** Reunión de Planificación del Sprint.



**Participantes:** Arquitecto de la Información (R), Equipo de Desarrollo.

ID	Información de Entrada	Fuente
P23	Informe de las tareas de desarrollo de los requisitos del Sprint	Tarea 3.4.1
E9	Técnicas ágiles para integrar la Arquitectura de la Información	Externa

ID	Productos de Salida	Destino
P25	Informe de las técnicas ágiles de la Arquitectura de la Información relacionadas con las tareas de desarrollo	Tarea 4.1.1, Tarea 4.3.1

#### **Tarea 3.4.3. Estimar duración de tareas**

**Descripción:** Estimar y calcular la duración de cada una de las tareas de desarrollo específicas.

**Técnicas:** Reunión de Planificación del Sprint.

**Participantes:** Equipo de Desarrollo (R), Arquitecto de la Información.

ID	Información de Entrada	Fuente
P23	Informe de las tareas de desarrollo de los requisitos del Sprint	Tarea 3.4.1

ID	Productos de Salida	Destino
P26	Informe sobre la estimación de las tareas de desarrollo	Tarea 3.5.1

#### **Tarea 3.4.4. Asignar responsables de tareas**

**Descripción:** Asignar un responsable perteneciente al Equipo de Desarrollo para cada una de las tareas de desarrollo especificadas.

**Técnicas:** Reunión de Planificación del Sprint.

**Participantes:** Equipo de Desarrollo (R).

ID	Información de Entrada	Fuente
P23	Informe de las tareas de desarrollo de los requisitos del Sprint	Tarea 3.4.1

P19	Informe sobre la disponibilidad del Equipo de Desarrollo para el Sprint	Tarea 3.1.2
-----	---	-------------

ID	Productos de Salida	Destino
P27	Informe de los responsables de las tareas de desarrollo	Tarea 3.5.1, Tarea 4.1.1

### Actividad 3.5. Planificación de las tareas de desarrollo

**Descripción:** En esta actividad, se debe realizar una estimación del esfuerzo diario durante el Sprint de las tareas de desarrollo (definidas en la Actividad 3.4).

Esta actividad sólo tiene la Tarea 3.5.1, que tiene como objetivo identificar y determinar los días en que se realizarán las tareas de desarrollo. En la Tarea 3.5.1, el Equipo de Desarrollo es el responsable de llevarla a cabo. No obstante, el Arquitecto de la Información también participa, con el fin de asegurar que la organización de las tareas permita implementarlas de manera incremental, dirigida por los niveles de fidelidad de los entregables de la AI. Además, la Tarea 3.5.1 utiliza como fuentes de información el Informe de las tareas de desarrollo de los requisitos del Sprint (P23), proporcionado por la Tarea 3.4.1, el Informe sobre la estimación de las tareas de desarrollo (P26), proporcionado por la Tarea 3.4.3, y el Informe de los responsables de las tareas de desarrollo (P27), proporcionado por la Tarea 3.4.4. El principal resultado de la Tarea 3.5.1 es un Diagrama de la planificación diaria de las tareas de desarrollo (P28).

A continuación, se detalla la tarea necesaria para la realización de la actividad:

#### Tarea 3.5.1. Determinar la planificación de las tareas

**Descripción:** Determinar el esfuerzo estimado para cada tarea de desarrollo.

**Técnicas:** Reunión de Planificación del Sprint.

**Participantes:** Equipo de Desarrollo, el Arquitecto de la Información.

ID	Información de Entrada	Fuente
P23	Informe de las tareas de desarrollo de los requisitos del Sprint	Tarea 3.4.1
P26	Informe sobre la estimación de las tareas de desarrollo	Tarea 3.4.3
P27	Informe de los responsables de las tareas de desarrollo	Tarea 3.4.4

ID	Productos de Salida	Destino
P28	Diagrama de la planificación diaria de las tareas de desarrollo	Tarea 4.1.1

#### 3.2.3.4. Grupo de Actividades de Inspección y Mejora Continua

Este grupo de actividades tiene como objetivo examinar y evaluar las tareas de desarrollo y las prácticas de trabajo utilizadas en el Sprint, con el fin de fomentar el DCU y la participación de los usuarios finales. Las actividades para este grupo corresponden a:

##### Actividad 4.1 Inspección de tareas de desarrollo

**Descripción:** En esta actividad, se debe examinar el estado de las tareas (definidas en la actividad 3.4 y planificadas en la actividad 3.5) que el Equipo de Desarrollo realiza durante la Ejecución del Sprint, con el fin de identificar inconvenientes y verificar el cumplimiento de las prioridades de los usuarios finales.

Esta actividad sólo tiene la Tarea 4.1.1, que trata de revisar el estado actual de las tareas y los problemas que tienen los miembros del Equipo de Desarrollo para llevarlas cabo. En la Tarea 4.1.1, el Equipo de Desarrollo es el responsable, pero el Arquitecto de la Información también participa, con el fin de inspeccionar las tareas y realizar un seguimiento del cumplimiento de los criterios de aceptación de los usuarios finales. Para lograr el objetivo de la Tarea 4.1.1, los participantes utilizan la reunión del Scrum Diario.

Además, la Tarea 4.1.1 utiliza como fuentes de información la Lista del Producto con los elementos descritos en formato de Historia de Usuario (P13) de la Tarea 2.2.2, el Informe de las tareas de desarrollo de los requisitos del Sprint (P23), proporcionado por la Tarea 3.4.1, el Informe de los responsables de las tareas de desarrollo (P27), proporcionado por la Tarea 3.4.4, el Informe de las técnicas ágiles de la Arquitectura de la Información relacionadas con las tareas de desarrollo (P25) de la Tarea 3.4.2, y el Diagrama de la planificación diaria de las tareas de desarrollo (P28) de la Tarea 3.5.1. Finalmente, el principal resultado de la Tarea 4.1.1 es el Informe de la revisión de las tareas de desarrollo (P29).

A continuación, se detalla la tarea necesaria para la realización de la actividad:

##### Tarea 4.1.1. Examinar las tareas de desarrollo

**Descripción:** Revisar el estado actual de las tareas de desarrollo y los desafíos que enfrentan los miembros del Equipo de Desarrollo para llevarlas a cabo.

**Técnicas:** Scrum Diario.

**Participantes:** Equipo de Desarrollo (R), Arquitecto de la Información, Scrum-UIA Master.

ID	Información de Entrada	Fuente
P13	Lista del Producto con los elementos descritos en formato de Historia de Usuario	Tarea 2.2.2
P23	Informe de las tareas de desarrollo de los requisitos del Sprint	Tarea 3.4.1
P27	Informe de los responsables de las tareas de desarrollo	Tarea 3.4.4
P28	Diagrama de la planificación diaria de las tareas de desarrollo	Tarea 3.5.1

ID	Productos de Salida	Destino
P29	Informe de la revisión de las tareas de desarrollo	Tarea 4.2.1

#### Actividad 4.2 Revisión del incremento del producto

**Descripción:** En esta actividad se debe evaluar el incremento del producto (que considera la definición de “Terminado”, especificado en la actividad 2.3) a través de la participación directa de los usuarios finales.

Esta actividad sólo tiene la Tarea 4.2.1, que tiene como objetivo evaluar el incremento potencial del producto, generado por el Equipo de Desarrollo durante la Ejecución del Sprint. En la Tarea 4.2.1, participan todos los miembros del equipo Scrum-UIA y se incluye a los usuarios finales para evaluar el incremento a través de su participación directa. Para lograr el objetivo de la Tarea 4.2.1, los participantes utilizan la reunión de la Revisión del Sprint.

Asimismo, en la Tarea 4.2.1, se utilizan como fuentes de información la Lista del Producto con los elementos descritos en formato de Historia de Usuario (P13) de la Tarea 2.2.2, el Informe de las tareas de desarrollo de los requisitos del Sprint (P23), proporcionado por la Tarea 3.4.1, y el Informe de la revisión de las tareas de desarrollo (P29) de la Tarea 4.1.1. Finalmente, el principal resultado de la Tarea 4.2.1 es el Informe sobre la revisión del incremento del producto (P30).

A continuación, se detalla la tarea necesaria para la realización de la actividad:

##### Tarea 4.2.1. Examinar el incremento potencial del producto

**Descripción:** Evaluar el incremento potencial del producto con la participación directa de los usuarios finales.

**Técnicas:** Revisión del Sprint.

**Participantes:** Equipo de Desarrollo (R), Arquitecto de la Información, Scrum-UIA Master, Dueño del Producto, Usuarios Finales.

ID	Información de Entrada	Fuente
P13	Lista del Producto con los elementos descritos en formato de Historia de Usuario	Tarea 2.2.2
P23	Informe de las tareas de desarrollo de los requisitos del Sprint	Tarea 3.4.1
P29	Informe de la revisión de las tareas de desarrollo	Tarea 4.1.1

ID	Productos de Salida	Destino
P30	Informe sobre la revisión del incremento del producto	Tarea 4.3.1

#### Actividad 4.3 Inspección de prácticas de trabajo

**Descripción:** En esta actividad se debe analizar el Sprint completado, con el fin de indagar mejoras para el siguiente Sprint.

Esta actividad sólo tiene la Tarea 4.3.1, que pretende indagar y crear un plan de mejoras para el siguiente Sprint. En la Tarea 4.3.1, se utiliza la reunión de la Retrospectiva del Sprint, donde participan todos los miembros del equipo Scrum-UIA. El Arquitecto de la Información revisa el desempeño de las prácticas y de las técnicas utilizadas en la integración de la AI, con el fin de identificar oportunidades de mejora, relacionadas con los usuarios finales.

En la Tarea 4.3.1, se utilizan como fuentes de información el Informe de las técnicas ágiles de la Arquitectura de la Información relacionadas con las tareas de desarrollo (P25) de la Tarea 3.4.2 y el Informe sobre la revisión del incremento del producto (P30) de la Tarea 4.2.1. Finalmente, el principal resultado de la Tarea 4.3.1 es el Informe sobre la inspección de las prácticas de trabajo (P31).

A continuación, se detalla la tarea necesaria para la realización de la actividad:

##### Tarea 4.3.1. Examinar el Sprint completado

**Descripción:** Inspeccionar el Sprint completado para averiguar las mejoras que se podrían introducir en el siguiente Sprint.

**Técnicas:** Retrospectiva del Sprint.

**Participantes:** Equipo de Desarrollo (R), Arquitecto de la Información, Scrum-UIA Master, Dueño del Producto.

ID	Información de Entrada	Fuente
P25	Informe de las técnicas ágiles de la Arquitectura de la Información relacionadas con las tareas de desarrollo	Tarea 3.4.2
P30	Informe sobre la revisión del incremento del producto	Tarea 4.2.1

ID	Productos de Salida
P31	Informe sobre la inspección de las prácticas de trabajo

### 3.2.4. Productos

#### 3.2.4.1. Descripción

Los niveles de especificación de los productos evolucionan a través de las diferentes tareas y actividades propuestas del modelo de la metodología Scrum-UIA. De este modo, las actividades iniciales del modelo producen diferentes fuentes de información que permiten, en tareas y actividades posteriores, generar información más detallada y concreta del modelo. Por ejemplo, los informes sobre las necesidades de los usuarios finales y de las prioridades del contenido, producidos en actividades iniciales, sirven de base para identificar y priorizar los requisitos del producto en actividades avanzadas del modelo.

Se ha elaborado la Figura 16 para mostrar las relaciones entre los productos, generados en las actividades del modelo de la metodología Scrum-UIA. Como se puede observar en la Figura 16, los productos identifican las interdependencias y las actividades donde se generan. Además, la dirección de las flechas de los productos señala cómo evoluciona el producto o la dependencia entre estos. De este modo, un producto inicial (flecha de inicio) puede corresponder a la fuente de información de entrada o de base del producto de destino (al final de la flecha), o esta relación puede señalar que el producto de origen contiene información pretendida por el producto de destino. Por ejemplo, los productos P1, P2, P3, P4, P5, P6 y P7 de las actividades iniciales del Grupo de Actividades de la Investigación Contextual corresponden a las fuentes de información de base para generar, en actividades posteriores, el panorama general del proyecto.

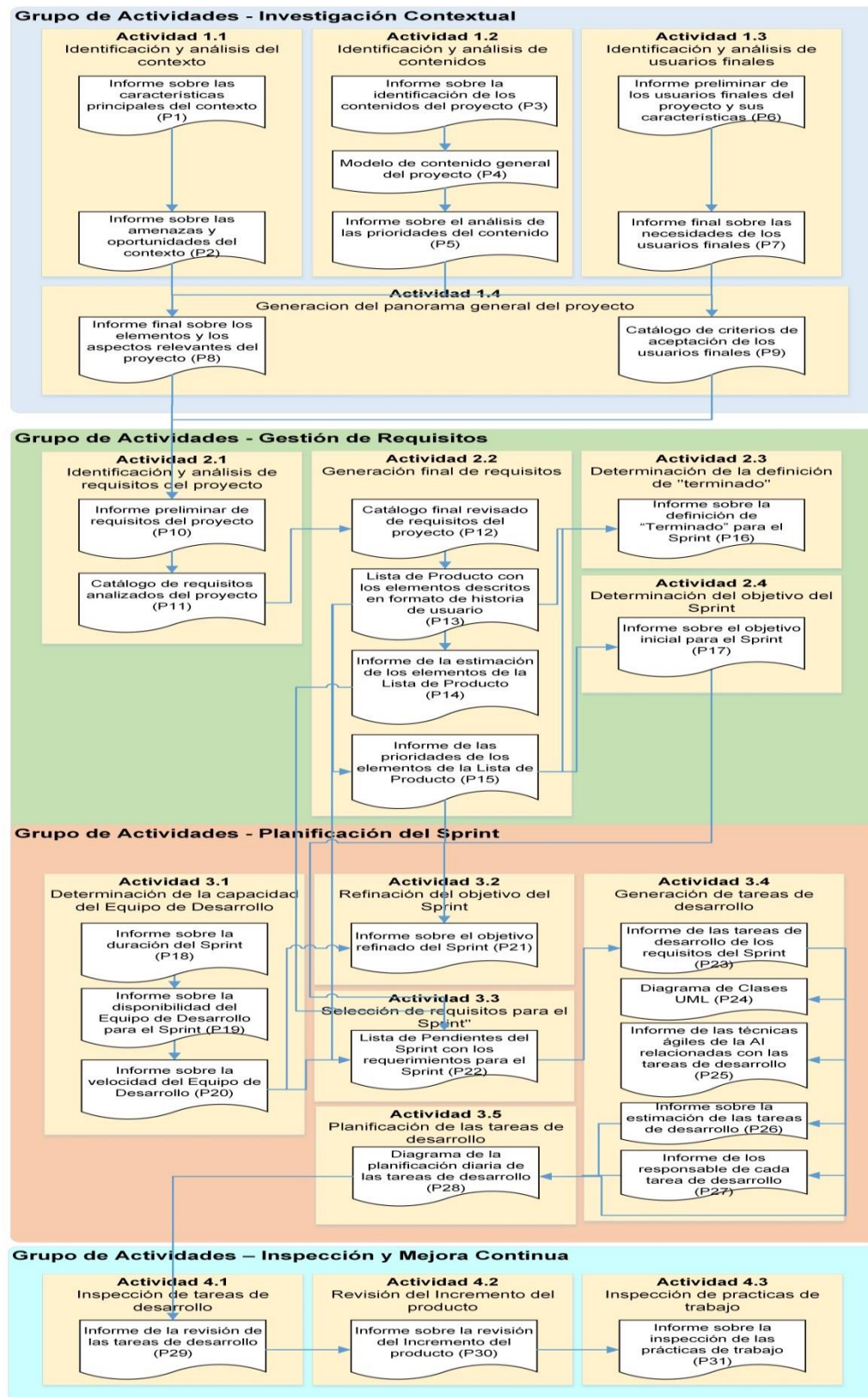


Figura 16: Relación entre los productos del modelo de la metodología Scrum-UIA.

#### 3.2.4.2. Productos

##### P1. Informe sobre las características principales del contexto

Este producto consiste en un registro de las características tanto internas como externas del entorno donde ocurre el proyecto. Para cada característica identificada, se deberá especificar lo siguiente:

- **Identificador de la característica del contexto:** Adoptando la sigla CCx, donde x corresponde al número correlativo y único, asignado a cada una de las características del contexto.
- **Nombre:** Nombre de la característica del contexto.

A continuación, se presenta un ejemplo concreto del producto “Informe sobre las características principales del contexto”:

P1. Informe sobre las características principales del contexto	
Autor: Arquitecto de la Información (L.R.)	
Fecha: 27 Julio 2016	
CC1:	Se ha incrementado la participación de la universidad en los estudiantes de entrada.
CC2:	La competencia de otras ofertas educativas superiores es alta.
CC3:	Se ha diversificado el número de titulaciones ofertadas.
CC4:	Se ha incrementado el número de estudiantes de postgrado.
CC5:	Los estudiantes y trabajadores provienen de diferentes culturas.

**Producto 1:** Ejemplo de Informe sobre las características principales del contexto (P1).



## P2. Informe sobre las amenazas y oportunidades del contexto

Producto que enumera las amenazas y oportunidades del contexto, que son necesarias a considerar en el tratamiento del SI. Para cada amenaza u oportunidad del contexto identificada, el producto debe especificar la siguiente información:

- **Identificador de la amenaza del contexto:** Adoptando la sigla **ACx**, donde x corresponde al número correlativo y único, asignado para las amenazas del contexto.
- **Identificador de la oportunidad del contexto:** Adoptando la sigla **OCx**, donde x corresponde al número correlativo y único, asignado para las oportunidades del contexto.
- **Nombre:** Nombre de la amenaza u oportunidad del contexto.

A continuación, se presenta un ejemplo concreto del producto “Informe sobre las amenazas y oportunidades del contexto”:

<b>P2. Informe sobre las amenazas y oportunidades del contexto</b>	
<b>Autor:</b> Arquitecto de la Información (L.R.)	
<b>Fecha:</b> 27 Julio 2016	
<b>Amenazas del contexto</b>	
<b>A1:</b> La competencia de otras universidades e instituciones próximas.	
<b>A2:</b> La disminución de la población en edad académica y considerable emigración de estudiantes a otras universidades.	
<b>A3:</b> La falta de presupuesto y financiamiento de proyectos específicos.	
<b>Oportunidades del contexto</b>	
<b>O1:</b> El uso de las nuevas tecnologías ofrece unas posibilidades de proyección y gestión del aprendizaje aún no suficientemente explotadas.	
<b>O2:</b> Entorno dinámico y complejo que obliga al cambio continuo.	
<b>O3:</b> Impacto de las tecnologías en las prácticas educativas.	
<b>O4:</b> Desarrollo de las nuevas tecnologías.	
<b>O5:</b> Elevada tasa de inmigración.	
<b>O6:</b> Mayor necesidad de formación continuada de los egresados.	

**Producto 2:** Ejemplo de Informe sobre las amenazas y oportunidades del contexto (P2).

### P3. Informe sobre la identificación de los contenidos del proyecto

Producto que enumera los contenidos que están asociados al proyecto. Para cada contenido identificado del proyecto, el producto debe indicar los siguientes datos:

- **Identificador del contenido identificado del proyecto:** Adoptando la sigla **CIPx**, donde x corresponde al número correlativo y único, asignado para cada contenido identificado del proyecto.
- **Nombre:** Nombre del contenido.
- **Descripción:** Descripción del contenido.
- **Ubicación:** Lugar físico o virtual donde se encuentra el contenido.

A continuación, se presenta un ejemplo concreto del producto “Informe sobre la identificación de los contenidos del proyecto”:

<b>P3. Informe sobre la identificación de los contenidos del proyecto</b>	
<b>Autor:</b> Arquitecto de la Información (L.R.)	
<b>Fecha:</b> 27 Julio 2016	
<b>CIP1:</b> Datos personales de los estudiantes.	
<b>Descripción:</b> Descripción detallada de los datos personales de los estudiantes de la universidad.	
<b>Ubicación:</b> C:\FileStudent\Dapees.xlsx	
<b>CIP2:</b> Catálogo de los cursos ofrecidos.	
<b>Descripción:</b> Identificación y descripción de los cursos ofrecidos por la universidad.	
<b>Ubicación:</b> C:\FileStudent\Cacuof.xlsx	
<b>CIP3:</b> Fotografías de los estudiantes e instructores.	
<b>Descripción:</b> Carpeta con fotografías en forma JPG de los estudiantes e instructores de la universidad.	
<b>Ubicación:</b> C:\File\picturesStudentsinstructos	

**Producto 3:** Ejemplo de Informe sobre la identificación de los contenidos del proyecto (P3).

#### **P4. Modelo de contenido general del proyecto**

Este producto representa, de manera gráfica, las relaciones de los contenidos del proyecto. Para cada modelo de contenido, se deben especificar los siguientes datos:

- **Identificador del modelo de contenido:** Adoptando la sigla **MCx**, donde x corresponde al número correlativo y único, asignado para cada modelo de contenido general del proyecto.
- **Nombre:** Nombre del modelo de contenido.
- **Descripción:** Descripción del modelo de contenido.
- **Modelo de contenido:** Representación visual de las relaciones coherentes del conjunto de los elementos del contenido y las conexiones lógicas entre ellas.

A continuación, se presenta un ejemplo concreto del producto “Modelo de contenido general del proyecto”:

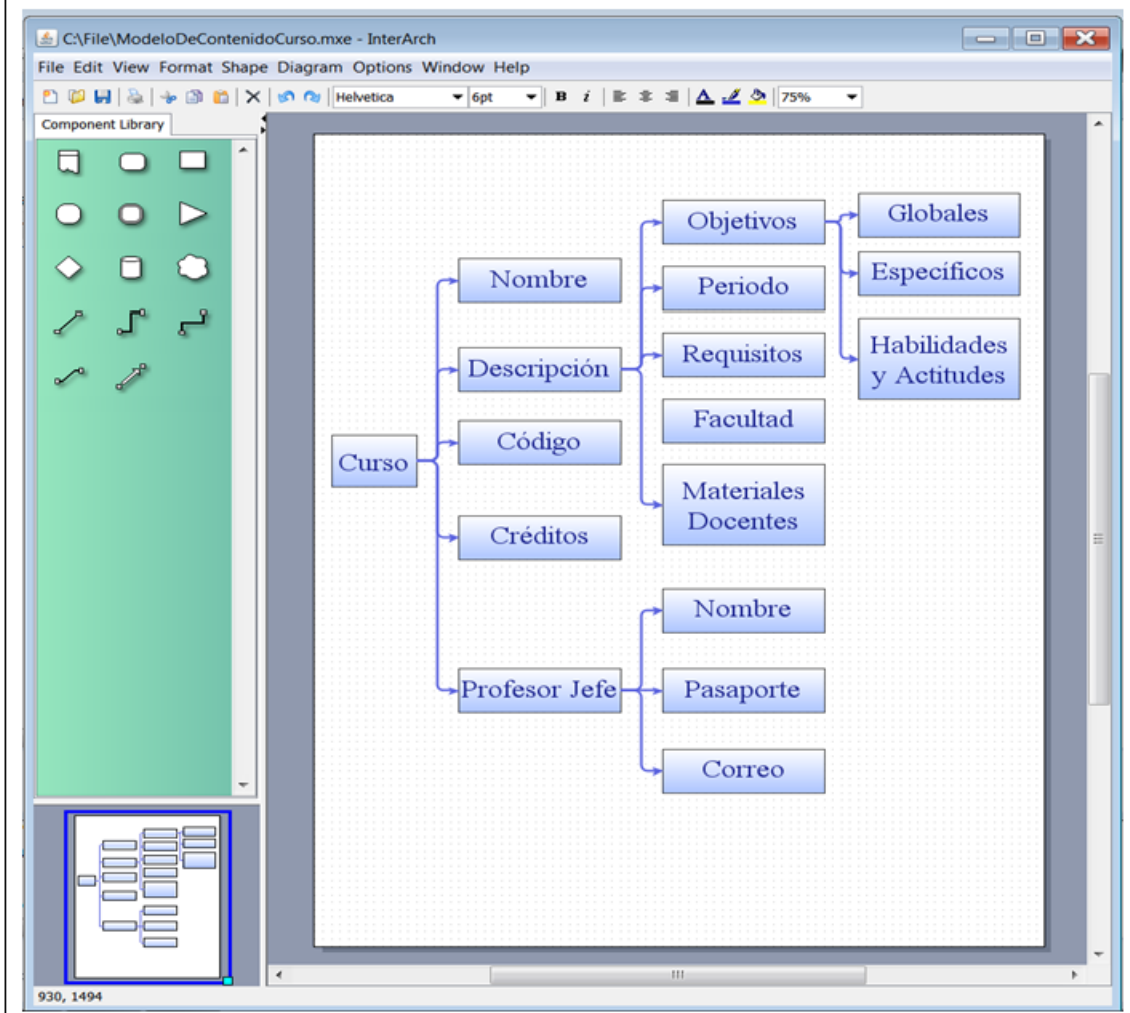
#### P4. Modelo de contenido general del proyecto

**Autor:** Arquitecto de la Información (L.R.)

**Fecha:** 27 Julio 2016

**MC1:** Modelo de contenido de los elementos relacionados con los cursos.

**Descripción:** Se describen los elementos de contenido de los que se componen los cursos ofrecidos por la universidad.



**Producto 4:** Ejemplo de Modelo de contenido general del proyecto (P4).

### P5. Informe sobre el análisis de las prioridades del contenido

Este producto representa las prioridades de los contenidos que se deben considerar en la elaboración del proyecto. Para cada ítem, el producto debe indicar lo siguiente:

- **Identificador del Contenido:** Adoptando la sigla **CPx**, donde x corresponde al número correlativo y único, asignado para cada contenido del proyecto.
- **Nombre:** Nombre del contenido.
- **Prioridad:** Número que representa la prioridad del contenido. Por ejemplo, 14, 20, 40, 13, siendo 40 la más alta prioridad.
- **Descripción:** Detalles del contenido.

A continuación, se presenta un ejemplo concreto del producto “Informe sobre el análisis de las prioridades del contenido”:

P5. Informe sobre el análisis de las prioridades del contenido	
<b>Autor:</b> Arquitecto de la Información (L.R.)	
<b>Fecha:</b> 27 Julio 2016	
<b>CP1:</b> Contenido de las entidades de conocimiento.	
<b>Descripción:</b> Información relacionada con el contenido de las entidades de conocimiento (cursos, módulos, actividades).	
<b>Prioridad:</b> 25.	
<b>CP2:</b> Descripción de las entidades de conocimiento.	
<b>Descripción:</b> Datos relacionados con la descripción de las entidades de conocimiento.	
<b>Prioridad:</b> 20.	
<b>CP3:</b> Calificaciones de las actividades de evaluación.	
<b>Descripción:</b> Valoraciones obtenidas por los estudiantes en las actividades de evaluación.	
<b>Prioridad:</b> 15.	

**Producto 5:** Ejemplo de Informe sobre el análisis de las prioridades del contenido (P5).

## **P6. Informe preliminar de los usuarios finales del proyecto y sus características**

Documento que enumera a los usuarios finales del proyecto. Para cada usuario final, el producto debe especificar lo siguiente:

- **Identificador del usuario final:** Adoptando la sigla **UFx**, donde x corresponde al número correlativo y único, asignado para cada uno de los usuarios finales.
- **Nombre:** Nombre del usuario final.
- **Descripción:** Detalles de las características del usuario final.

A continuación, se presenta un ejemplo concreto del producto “Informe preliminar de los usuarios finales del proyecto y sus características”:

<b>P6. Informe preliminar de los usuarios finales del proyecto y sus características</b> <b>Autor:</b> Arquitecto de la Información (L.R.) <b>Fecha:</b> 27 Julio 2016
<b>UF31:</b> Estudiante. <b>Descripción:</b> Rango de edad entre 17-55 años. Demanda información sobre las entidades de conocimiento y retroalimentación de las evaluaciones. Conocimiento en informática intermedio-alto.
<b>UF32:</b> Instructor. <b>Descripción:</b> Rango de edad entre 22-65 años. Demanda información sobre la gestión de las asignaturas. Conocimiento en informática intermedio.
<b>UF33:</b> Administrador. <b>Descripción:</b> Rango de edad entre 20-60 años. Demanda información sobre la matrícula de los estudiantes y gestión de las asignaturas. Conocimiento en informática intermedio-bajo.
<b>UF34:</b> Observador. <b>Descripción:</b> Rango de edad entre 35-70 años. Demanda información sobre el progreso de los estudiantes. Conocimiento en informática intermedio-bajo.

**Producto 6:** Ejemplo de Informe preliminar de los usuarios finales del proyecto y sus características (P6).

### P7. Informe final sobre las necesidades de los usuarios finales

Este producto registra las necesidades de los usuarios finales, detallados en el “Informe preliminar de los usuarios finales del proyecto y sus características” (P6). Para cada usuario final, el producto debe especificar los siguientes datos:

- **Identificador de la necesidad del usuario final:** Adoptando la sigla **NUFx.y**, donde  $x$  es el identificador del usuario final, definido en el producto “Informe preliminar de los usuarios finales del proyecto y sus características” (P6), mientras que  $y$  corresponde al número correlativo y único, asignado para cada necesidad de los usuarios finales. Por ejemplo, para el primer usuario final y su tercera necesidad establecida, el identificador se especifica como NUF1.3.
- **Usuario final:** Nombre del usuario final.
- **Necesidad:** Necesidad del usuario final.
- **Descripción:** Detalles de la necesidad del usuario final.

A continuación, se presenta un ejemplo concreto del producto “Informe final sobre las necesidades de los usuarios finales”:

P7. Informe final sobre las necesidades de los usuarios finales	
<b>Autor:</b> Arquitecto de la Información (L.R.)	
<b>Fecha:</b> 28 Julio 2016	
NUF31.1: Estudiante.	
<b>Necesidad:</b> Revisar la descripción y el contenido de las entidades de conocimientos (cursos, grupos, módulos).	
<b>Descripción:</b> Estudiantes requieren consultar el detalle y los requisitos de cada entidad de conocimiento.	
NUF31.2: Estudiante.	
<b>Necesidad:</b> Consultar las actividades de evaluación.	
<b>Descripción:</b> Estudiantes demandan revisar el proceso de evaluación y obtener detalles de la retroalimentación de las actividades de evaluación.	
NUF31.3: Estudiante.	
<b>Necesidad:</b> Gestionar las fuentes de información.	
<b>Descripción:</b> Estudiantes requieren manipular los archivos utilizados y generados en cada entidad de conocimiento mediante un repositorio centralizado.	

**Producto 7:** Ejemplo de Informe final sobre las necesidades de los usuarios finales (P7).

## P8. Informe final sobre los elementos y los aspectos relevantes del proyecto

Este producto consiste en un registro de las prioridades del proyecto que permiten identificar el nivel de importancia de los requisitos en el proceso de priorización. Para cada ítem, el producto debe especificar los siguientes datos:

- **Identificador del elemento del aspecto:** Adoptando la sigla **AEx.y**, donde *x* es el identificador del aspecto relevante del proyecto, mientras que *y* corresponde al número correlativo y único, asignado para cada elemento identificado. Por ejemplo, para el primer aspecto relevante y su segundo elemento establecido, el identificador se especifica como AE1.2.
- **Aspecto:** Nombre del aspecto relevante del proyecto.
- **Elemento:** Nombre del elemento del aspecto.
- **Prioridad del elemento:** Prioridad del elemento, utilizando la escala cualitativa *Alta*, *Media* y *Baja*.
- **Prioridad del aspecto:** Prioridad del aspecto, utilizando la escala cualitativa *Alta*, *Media* y *Baja*.

A continuación, se presenta un ejemplo concreto del producto “Informe final sobre los elementos y los aspectos relevantes del proyecto”:

P8. Informe final sobre los elementos y los aspectos relevantes del proyecto		
Autor: Arquitecto de la Información (L.R.)		
Fecha: 28 Julio 2016		
Identificador	Aspectos relevantes del SI de gestión de aprendizaje	Prioridad
12	Usabilidad	Alta
13	Contenido	Media
14	Valor de Negocio	Baja
<b>Elementos del aspecto relevante de usabilidad</b>		
AE12.1: Proveer adaptabilidad al entorno de aprendizaje.		
<b>Prioridad:</b> Alta.		
AE12.2: Proveer facilidad de aprendizaje.		
<b>Prioridad:</b> Alta.		
AE12.3: Proveer accesibilidad a la información (opciones de formato).		
<b>Prioridad:</b> Alta.		
AE12.4: Reducir la carga mental de trabajo.		
<b>Prioridad:</b> Media.		

**Producto 8:** Ejemplo de Informe final sobre los elementos y los aspectos relevantes del proyecto (P8).



### P9. Catálogo de criterios de aceptación de los usuarios finales

Este producto consiste en un registro de los criterios de aceptación de los usuarios finales, detallados en el “Informe preliminar de los usuarios finales del proyecto y sus características” (P6). Para cada criterio de aceptación, el producto debe especificar los siguientes datos:

- **Identificador del criterio de aceptación del usuario final:** Adoptando la sigla **CAUF $x$ . $y$** , donde  $x$  es el identificador del usuario final, definido en el producto “Informe preliminar de los usuarios finales del proyecto y sus características” (P6), mientras que  $y$  corresponde al número correlativo y único, asignado para cada criterio de aceptación de los usuarios finales. Por ejemplo, para el segundo usuario final y su quinto criterio de aceptación identificado, el identificador se especifica como CAUF2.5.
- **Usuario final:** Nombre del usuario final.
- **Nombre:** Nombre de criterio de aceptación del usuario final.
- **Descripción:** Detalles del criterio de aceptación del usuario final.

A continuación, se presenta un ejemplo concreto del producto “Catálogo de criterios de aceptación de los usuarios finales”:

P9. Catálogo de criterios de aceptación de los usuarios finales	
<b>Autor:</b> Arquitecto de la Información (L.R.)	
<b>Fecha:</b> 28 Julio 2016	
CAUF31.1: Comprobar las acciones.	
<b>Usuario Final:</b> Estudiante.	
<b>Descripción:</b> Permitir comprobar cómo las acciones afectan a la salida del SI.	
CAUF31.2: Interfaz adaptable.	
<b>Usuario Final:</b> Estudiante.	
<b>Descripción:</b> Mostrar opciones para personalizar la interfaz, de acuerdo al perfil del usuario.	
CAUF31.3: Presentación de la información.	
<b>Usuario Final:</b> Estudiante.	
<b>Descripción:</b> Mostrar alternativas gráficas en la presentación de la información.	

**Producto 9:** Ejemplo de Catálogo de criterios de aceptación de los usuarios finales (P9).

### P10. Informe preliminar de requisitos del proyecto

Producto que enumera los requisitos del proyecto. Para cada requisito, el producto debe indicar la siguiente información:

- **Identificador del requisito:** Adoptando la sigla **Rx**, donde x corresponde al número correlativo y único, asignado para cada requisito identificado del proyecto.
- **Nombre:** Nombre del requisito.
- **Descripción:** Descripción del requisito.

A continuación, se presenta un ejemplo concreto del producto “Informe preliminar de requisitos del proyecto”:

<b>P10. Informe preliminar de requisitos del proyecto</b>	
<b>Autor:</b> Dueño del Producto (J.P.)	
<b>Fecha:</b> 28 Julio 2016	
<b>R1:</b> Crear y editar cursos	
<b>Descripción:</b> Crear y editar los cursos ofrecidos por la universidad, con el fin de configurar sus datos generales (por ejemplo, nombre, objetivos, requisitos, profesores).	
<b>R2:</b> Gestionar cursos en un entorno integrado	
<b>Descripción:</b> Proveer un entorno virtual que permite consultar y gestionar los diferentes cursos que se llevan a cabo.	
<b>R3:</b> Crear y editar actividades de evaluación	
<b>Descripción:</b> Generar actividades de evaluación, asociadas a los cursos impartidos por la universidad, con el fin de permitir su gestión electrónica en todas sus fases.	
<b>R4:</b> Configurar perfiles de estudiantes y profesores	
<b>Descripción:</b> Configurar datos personales y preferencias de los estudiantes y profesores.	

**Producto 10:** Ejemplo de Informe preliminar de requisitos del proyecto (P10).

### P11. Catálogo de requisitos analizados del proyecto

Este producto corresponde al análisis de los requisitos del proyecto, detallados en el “Informe preliminar de requisitos del proyecto” (P10), donde se identifican los usuarios finales, relacionados con los requisitos y el valor de negocio que se persiguen. Para cada requisito, el producto debe indicar la siguiente información:

- **Identificador del requisito:** Utilizando el acrónimo **Rx**, definido en el producto P10 (Informe preliminar de requisitos del proyecto).
- **Nombre:** Nombre del requisito.
- **Descripción:** Descripción del requisito.
- **Usuarios finales:** Lista de los usuarios finales, que están relacionados con el requisito.
- **Análisis:** Descripción del análisis del requisito.

A continuación, se presenta un ejemplo concreto del producto “Catálogo de requisitos analizados del proyecto”:

<b>P11. Catálogo de requisitos analizados del proyecto</b>	
<b>Autor:</b> Dueño del Producto (J.P.)	
<b>Fecha:</b> 28 Julio 2016	
<b>R1:</b> Crear y editar cursos.	
<b>Descripción:</b> Crear y editar los cursos ofrecidos por la universidad, con el fin de configurar sus datos generales (por ejemplo, nombre, objetivos, requisitos, profesores).	
<b>Usuarios finales:</b> Instructores, Administradores.	
<b>Análisis:</b> Este requisito permite definir y configurar las materias que forman los planes de estudio de la universidad.	
<b>R2:</b> Gestionar cursos en un entorno integrado.	
<b>Descripción:</b> Proveer un entorno virtual que permite consultar y gestionar los diferentes cursos que se llevan a cabo.	
<b>Usuarios finales:</b> Estudiantes.	
<b>Análisis:</b> Este requisito permite que la información de aprendizaje de los estudiantes se pueda organizar y configurar.	

**Producto 11:** Ejemplo de Catálogo de requisitos analizados del proyecto (P11).

## P12. Catálogo final revisado de requisitos del proyecto

Es el catálogo final de los requisitos del proyecto, que identifica las dependencias de los requisitos. Para cada requisito, el producto debe indicar los siguientes datos:

- **Identificador del requisito:** Utilizando el acrónimo **Rx**, definido en el producto P10 (Informe preliminar de requisitos del proyecto).
- **Nombre:** Nombre del requisito.
- **Dependencias:** Lista de los requisitos que dependen del requisito.
- **Usuarios finales:** Lista de los usuarios finales, que están relacionados con el requisito.
- **Análisis:** Descripción del análisis del requisito.
- **Descripción:** Descripción del requisito.

A continuación, se presenta un ejemplo concreto del producto “Catálogo final revisado de requisitos del proyecto”:

P12. Catálogo final revisado de requisitos del proyecto	
Autor: Dueño del Producto (J.P.)	
Fecha: 28 Julio 2016	
R1: Crear y editar cursos.	
Descripción: Crear y editar los cursos ofrecidos por la universidad, con el fin de configurar sus datos generales (por ejemplo, nombre, objetivos, requisitos, profesores).	
Usuarios finales: Instructores, Administradores.	
Análisis: Este requisito permite definir y configurar las materias que forman los planes de estudio de la universidad.	
R2: Gestionar cursos en un entorno integrado.	
Descripción: Proveer un entorno virtual, que permite consultar y gestionar los diferentes cursos que se llevan a cabo.	
Usuarios finales: Estudiantes.	
Análisis: Este requisito permite que la información de aprendizaje de los estudiantes se pueda organizar y configurar.	
Dependencias: R1.	

**Producto 12:** Ejemplo de Catálogo final revisado de requisitos del proyecto (P12).

### P13. Lista del Producto con los elementos descritos en formato de Historia de Usuario

Producto que enumera los requisitos del proyecto en formato de historia de usuario (es decir, cada requisito del SI identifica el usuario final, la necesidad que cubre y el valor de negocio que se persigue). Para cada requisito, el documento debe indicar lo siguiente:

- **Identificador del requisito:** Utilizando el acrónimo **Rx**, definido en el producto P10 (Informe preliminar de requisitos del proyecto).
- **Capacidad:** Función demandada del SI.
- **Valor de negocio:** Beneficio que se busca, en cuanto a la capacidad requerida.
- **Usuarios finales:** Lista de los usuarios finales, que están relacionados con el requisito.
- **Dependencias:** Lista de los requisitos que son dependientes del requisito.
- **Nombre:** Nombre del requisito.
- **Descripción:** Descripción del requisito.
- **Tipo:** Clasificación del requisito. Ejemplo, Historia de Usuario, *Historia de Usuario de la AI*, otros. *Historia de Usuario de la AI* corresponde a los requisitos relacionados con la AI, que permiten dirigir el desarrollo incremental mediante los niveles de fidelidad de los entregables de la AI.
- **Criterios de aceptación de los usuarios finales:** Listado de los juicios de aprobación de los usuarios finales.

A continuación, se presenta un ejemplo concreto del producto “Lista del Producto con los elementos descritos en formato de Historia de Usuario”:

<b>P13. Lista del Producto con los elementos descritos en formato de Historia de Usuario</b>
<b>Autor:</b> Dueño del Producto (J.P.)
<b>Fecha:</b> 29 Julio 2016
<b>R1:</b> Crear y editar cursos.  <b>Como</b> Instructores y Administradores, <b>Queremos</b> crear y editar los cursos, <b>con el fin de</b> definir y configurar las materias que forman los planes de estudio de la universidad.  <b>Descripción:</b> Crear y editar los cursos ofrecidos por la universidad, con el fin de configurar sus datos generales (por ejemplo, nombre, objetivos, requisitos, profesores).  <b>Criterios de aceptación de los usuarios finales:</b> <ul style="list-style-type: none"><li>• CAUF32.1: Acciones reversibles.</li><li>• CAUF33.1: Gestión de errores.</li></ul>

**Producto 13:** Ejemplo de Lista del Producto con los elementos descritos en formato de Historia de Usuario (P13).

#### P14. Informe de la estimación de los elementos de la Lista del Producto

Este producto corresponde a la estimación en puntos de historias y días ideales de los elementos de la Lista del Producto. Para cada ítem identificado, se deberá indicar lo siguiente:

- **Identificador del requisito:** Utilizando el acrónimo **Rx**, definido en el producto P10 (Informe preliminar de requisitos del proyecto).
- **Nombre:** Nombre del requisito.
- **Estimación total:** Valoración numérica del tiempo total, requerido para implementar el requisito.
- **Tipo de estimación total:** Tipo de estimación total del requisito. Ejemplo: Días, Horas o Minutos.
- **Puntos de historias:** Estimación numérica para expresar el tamaño global del requisito. Por ejemplo, un requisito estimado en cuatro puntos de historia indica que es el doble de grande, complejo o riesgoso que un requisito estimado con dos puntos de historia.
- **Estimación restante:** Valoración numérica del tiempo restante, requerido para implementar el requisito.
- **Tipo de estimación restante:** Tipo de estimación restante del requisito. Ejemplo: Días, Horas o Minutos.

A continuación, se presenta un ejemplo concreto del producto “Informe de la estimación de los elementos de la Lista del Producto”:

P14. Informe de la estimación de los elementos de la Lista del Producto	
Autor: Equipo de Desarrollo (M.R. y A.G.)	
Fecha: 29 Julio 2016	
R1: Crear y editar cursos	
Puntos de historias: 6	
Estimación total: 2 días	
Estimación restante: 2 días	
R2: Gestionar cursos en un entorno integrado	
Puntos de historias: 15	
Estimación total: 5 días	
Estimación restante: 5 días	
R3: Crear y editar actividades de evaluación	
Puntos de historias: 9	
Estimación total: 3 días	
Estimación restante: 3 días	

**Producto 14:** Ejemplo de Informe de la estimación de los elementos de la Lista del Producto (P14).

### **P15. Informe de las prioridades de los elementos de la Lista del Producto**

Este es el producto que enumera los requisitos del proyecto con las prioridades asignadas por el Arquitecto de la Información y el Dueño del Producto. Para cada requisito, el producto debe especificar los siguientes datos:

- **Identificador del requisito:** Utilizando el acrónimo **Rx**, definido en el producto P10 (Informe preliminar de requisitos del proyecto).
- **Nombre:** Nombre del requisito.
- **Prioridad AI:** Estimación numérica para indicar la prioridad del requisito, de acuerdo al Arquitecto de la Información. Por ejemplo, 10, 15, 5, 13, siendo 15 la prioridad más alta.
- **Prioridad DP:** Estimación numérica para indicar la prioridad del requisito, de acuerdo al Dueño del Producto. Por ejemplo, 20, 17, 11, 5, siendo 20 la prioridad más alta.
- **Elementos de los aspectos del proyecto:** Lista de los elementos de los aspectos relevantes del proyecto que se encuentran relacionados con el requisito, obtenidos desde el “Informe final sobre los elementos y los aspectos relevantes del proyecto” (P8). Estos elementos se utilizan en el método de priorización propuesto para Scrum-UIA, que permite conducir el proceso de priorización a través de las prioridades del proyecto (en la sección 3.3, se presentan los detalles de este método de priorización).
- **Clasificación QMPSR:** Estimación numérica para indicar la prioridad del requisito, de acuerdo a los elementos de los aspectos relevantes del proyecto (en la sección 3.3 se describe cómo calcular esta clasificación). Por ejemplo, 3, 7, 11, 4, siendo 11 la prioridad más alta.

A continuación, se introduce un ejemplo concreto del producto “Informe de las prioridades de los elementos de la Lista del Producto”:

<b>P15. Informe de las prioridades de los elementos de la Lista del Producto</b>		
<b>Autor:</b> Dueño del Producto (J.P.)		
<b>Fecha:</b> 29 Julio 2016		
<b>Identificador</b>	<b>Aspectos relevantes</b>	<b>Prioridad</b>
12	Usabilidad	Alta
13	Contenido	Media
14	Valor de Negocio	Baja
<b>R1:</b> Crear y editar cursos		
<b>Prioridad Arquitecto de la Información:</b> 50		
<b>Prioridad Dueño del Producto:</b> 45		
<b>Elementos de los aspectos asociados:</b>		
<ul style="list-style-type: none"><li>• <b>AE12.5:</b> Contemplar los potenciales errores</li><li>• <b>AE13.5:</b> Preferencias de los estudiantes</li><li>• <b>AE13.6:</b> Descripción de las entidades de conocimiento</li><li>• <b>AE14.4:</b> Gestionar las entidades de conocimiento</li><li>• <b>AE14.6:</b> Coordinar composición de las entidades de conocimiento</li></ul>		
<b>Clasificación QMPSR:</b> 10,83		

**Producto 15:** Ejemplo de Informe de las prioridades de los elementos de la Lista del Producto (P15).



### **P16. Informe sobre la definición de “Terminado” para el Sprint**

Este producto es la definición de “Terminado”, acordada por todos los integrantes del equipo. Para cada ítem, el producto debe indicar lo siguiente:

- **Identificador del Sprint:** Adoptando la sigla **Sx**, donde *x* corresponde al número correlativo y único, asignado para cada Sprint del proyecto.
- **Nombre:** Nombre del Sprint.
- **Descripción del Sprint:** Descripción del Sprint.
- **Definición de terminado:** Descripción de la definición de “Terminado” para el Sprint.

A continuación, se presenta un ejemplo concreto del producto “Informe sobre la definición de “Terminado” para el Sprint”:

<b>P16. Informe sobre la definición de “Terminado” para el Sprint</b> <b>Autor:</b> Dueño del Producto (J.P.) <b>Fecha:</b> 31 Julio 2016
<b>S1: Sprint 1</b>
<b>Descripción del Sprint:</b> Sprint inicial del SI de gestión de aprendizaje <b>Definición de “Terminado”:</b> <ul style="list-style-type: none"><li>• Las historias de usuarios deben cumplir con los criterios de aceptación de los usuarios finales</li><li>• El diseño de las pantallas debe cumplir con la estructura de las maquetas (Wireframes)</li><li>• Los datos proporcionados deben considerar alternativas de accesibilidad</li></ul>

**Producto 16:** Ejemplo de Informe sobre la definición de “Terminado” para el Sprint (P16).

### **P17. Informe sobre el objetivo inicial para el Sprint**

Producto que identifica el objetivo para el Sprint. Para cada ítem, el producto debe indicar lo siguiente:

- **Identificador del Sprint:** Utilizando el acrónimo Sx, definido en el producto P16 (Informe sobre la definición de “Terminado” para el Sprint).
- **Nombre:** Nombre del Sprint.
- **Objetivo:** Descripción del objetivo del Sprint.

A continuación, se presenta un ejemplo concreto del producto “Informe sobre el objetivo inicial para el Sprint”:

<b>P17. Informe sobre el objetivo inicial para el Sprint</b>	
<b>Autor:</b> Dueño del Producto (J.P.)	
<b>Fecha:</b> 31 Julio 2016	
<b>S1:</b> Sprint 1	
<b>Objetivo del Sprint:</b> Proveer las bases arquitectónicas (diseño estructural del contenido y modelo de navegación) del SI de gestión de aprendizaje.	

**Producto 17:** Ejemplo de Informe sobre el objetivo inicial para el Sprint (P17).

### **P18. Informe sobre la duración del Sprint**

Este producto identifica el periodo de duración que tendrá el Sprint. Para cada ítem, el producto debe indicar lo siguiente:

- **Identificador del Sprint:** Utilizando el acrónimo **Sx**, definido en el producto P16 (Informe sobre la definición de “Terminado” para el Sprint).
- **Nombre:** Nombre del Sprint.
- **Inicio:** Fecha de inicio del Sprint.
- **Fin:** Fecha de finalización del Sprint.

A continuación, se presenta un ejemplo concreto del producto “Informe sobre la duración del Sprint”:

<b>P18. Informe sobre la duración del Sprint</b>	
<b>Autor:</b> Equipo de Desarrollo (M.R. y A.G.)	
<b>Fecha:</b> 31 Julio 2016	
<b>S1:</b> Sprint 1	
<b>Inicio del Sprint:</b> 8 de Agosto del 2017	
<b>Fin del Sprint:</b> 19 de Agosto del 2017	

**Producto 18:** Ejemplo de Informe sobre la duración del Sprint (P18).

### **P19. Informe sobre la disponibilidad del Equipo de Desarrollo para el Sprint**

Este producto lista los recursos de los que el Equipo de Desarrollo dispondrá en cada uno de los Sprints del proyecto. Para cada Sprint, el producto debe indicar los siguientes datos:

- **Identificador del miembro del Equipo de Desarrollo:** Adoptando la sigla **MEDx**, donde *x* es número correlativo y único, asignado a cada miembro del Equipo de Desarrollo.
- **Nombre:** Nombre del miembro del Equipo de Desarrollo.
- **Identificador del Sprint:** Utilizando el acrónimo **Sx**, definido en el producto P16 (Informe sobre la definición de “Terminado” para el Sprint).
- **Nombre Sprint:** Nombre del Sprint.
- **Disponibilidad para el siguiente Sprint:** Especificación en días-persona de la disponibilidad del miembro del Equipo de Desarrollo para el siguiente Sprint.
- **Disponibilidad en el Sprint anterior:** Especificación en días-persona de la disponibilidad del miembro del Equipo de Desarrollo del Sprint anterior.

A continuación, se presenta un ejemplo concreto del producto “Informe sobre la disponibilidad del Equipo de Desarrollo para el Sprint”:

<b>P19. Informe sobre la disponibilidad del Equipo de Desarrollo para el Sprint</b>	
<b>Autor:</b> Equipo de Desarrollo (M.R. y A.G.)	
<b>Fecha:</b> 31 Julio 2016	
S1: Sprint 1	
MED1: Adams Galkus	
Disponibilidad para el siguiente Sprint: 5	
Disponibilidad en el Sprint anterior: 4	
MED2: Luis Rojas	
Disponibilidad para el siguiente Sprint: 6	
Disponibilidad en el Sprint anterior: 7	
MED3: Mario Reyes	
Disponibilidad para el siguiente Sprint: 4	
Disponibilidad en el Sprint anterior: 6	
Disponibilidad Total para el siguiente Sprint: 15	
Disponibilidad Total en el Sprint anterior: 17	

**Producto 19:** Ejemplo de Informe sobre la disponibilidad del Equipo de Desarrollo para el Sprint (P19).

## P20. Informe sobre la velocidad del Equipo de Desarrollo

Este producto identifica la estimación de la velocidad del Equipo de Desarrollo para el siguiente Sprint. Para cada ítem, el producto debe indicar los siguientes datos:

- **Identificador del Sprint:** Utilizando el acrónimo **Sx**, definido en el producto P16 (Informe sobre la definición de “Terminado” para el Sprint).
- **Nombre Sprint:** Nombre del Sprint.
- **Velocidad real:** Puntos de historias, completados por el Equipo de Desarrollo en el Sprint anterior, obtenidos desde la información externa E8.
- **Factor de dedicación:** Estimación, expresada en términos porcentuales, respecto al nivel de dedicación y concentración que tiene el Equipo de Desarrollo durante la ejecución de un Sprint. El factor de dedicación se obtiene mediante la proporción entre la *Velocidad Real* del Equipo de Desarrollo, adquirida en la ejecución del último Sprint (P20) y los días-persona que tuvieron disponibles (suma de la *Disponibilidad en el Sprint anterior* de todos los miembros del Equipo de desarrollo, obtenida desde el P19).
- **Velocidad estimada:** Velocidad del Equipo de Desarrollo, estimada en puntos de historias, que se obtiene mediante la razón entre el *Factor de dedicación* y la disponibilidad en días-persona del Equipo de Desarrollo para el siguiente Sprint (suma de la *Disponibilidad para el siguiente Sprint* de todos los miembros del Equipo de desarrollo, obtenida desde el producto P19, Informe sobre la disponibilidad del Equipo de Desarrollo para el Sprint).

A continuación, se presenta un ejemplo concreto del producto “Informe sobre la velocidad del Equipo de Desarrollo”:

P20. Informe sobre la velocidad del Equipo de Desarrollo	
Autor: Equipo de Desarrollo (M.R. y A.G.)	
Fecha: 31 Julio 2016	
S1: Sprint 1	
Velocidad real: 50	
Factor de dedicación: 2,9 [ <i>Velocidad Real / Total Disponibilidad del Sprint anterior (P19)</i> ]	
Velocidad estimada: 43,5 [ <i>Factor de dedicación * Total Disponibilidad para el siguiente Sprint (P19)</i> ]	

**Producto 20:** Ejemplo de Informe sobre la velocidad del Equipo de Desarrollo (P20).

### **P21. Informe sobre el objetivo refinado del Sprint**

Este producto identifica el objetivo que finalmente se utilizará para el siguiente Sprint. Para cada ítem, el producto debe indicar lo siguiente:

- **Identificador del Sprint:** Utilizando el acrónimo **Sx**, definido en el producto P16 (Informe sobre la definición de “Terminado” para el Sprint).
- **Objetivo:** Descripción final del objetivo del Sprint.

A continuación, se presenta un ejemplo concreto del producto “Informe sobre el objetivo refinado del Sprint”:

<b>P21. Informe sobre el objetivo refinado del Sprint</b>	
<b>Autor:</b> Dueño del Producto (J.P.)	
<b>Fecha:</b> 31 Julio 2016	
<b>S1:</b> Sprint 1	
<b>Objetivo del Sprint:</b> Proveer las bases arquitectónicas (diseño estructural del contenido y modelo de navegación) del SI de gestión de aprendizaje, para conducir el desarrollo de los siguientes Sprints.	

**Producto 21:** Ejemplo de Informe sobre el objetivo refinado del Sprint (P21).

## **P22. Lista de Pendientes del Sprint con los requisitos para el Sprint**

Este producto corresponde a un catálogo de los requisitos seleccionados, que se deben implementar en el Sprint. Para cada Sprint, el producto debe indicar lo siguiente:

- **Identificador del Sprint:** Utilizando el acrónimo **Sx**, definido en el producto P16 (Informe sobre la definición de “Terminado” para el Sprint).
- **Nombre Sprint:** Nombre del Sprint.
- **Requisitos:** Lista de requisitos seleccionados de la Lista del Producto.

A continuación, se presenta un ejemplo concreto del producto “Lista de Pendientes del Sprint con los requisitos para el Sprint”:

<b>P22. Lista de Pendientes del Sprint con los requisitos para el Sprint</b>	
<b>Autor:</b> Equipo de Desarrollo (M.R. y A.G.)	
<b>Fecha:</b> 31 Julio 2016	
<b>S1: Sprint 1</b>	
<b>Requisitos:</b> <ul style="list-style-type: none"><li>• <b>R1:</b> Crear y editar cursos</li><li>• <b>R2:</b> Gestionar cursos en un entorno integrado</li><li>• <b>R4:</b> Configurar perfiles de estudiantes y profesores</li></ul>	
<b>Total de Puntos de Historias:</b> 33	
<b>Total de Días-Persona:</b> 11	
<b>Total de Requisitos:</b> 3	

**Producto 22:** Ejemplo de Lista de Pendientes del Sprint con los requisitos para el Sprint (P22).

### **P23. Informe de las tareas de desarrollo de los requisitos del Sprint**

Es un catálogo con las tareas de desarrollo, definidas para cada uno de los requisitos seleccionados para el Sprint. Para cada tarea de desarrollo, el producto debe indicar los siguientes datos:

- **Identificador de la tarea de desarrollo:** Adoptando la sigla **TD $x$ . $y$** , donde  $x$  es el identificador del requisito, establecido en el producto “Informe preliminar de requisitos del proyecto” (P10), mientras que  $y$  corresponde al número correlativo y único, asignado para cada tarea de desarrollo. Por ejemplo, para el primer requisito y su segunda tarea de desarrollo establecida, el identificador se especifica como TD1.2.
- **Nombre:** Nombre de la tarea de desarrollo.
- **Nombre requisito:** Nombre del requisito que está asociado a la tarea de desarrollo.

A continuación, se presenta un ejemplo concreto del producto “Informe de las tareas de desarrollo de los requisitos del Sprint”:

<b>P23. Informe de las tareas de desarrollo de los requisitos del Sprint</b>	
<b>Autor:</b> Equipo de Desarrollo (M.R. y A.G.)	
<b>Fecha:</b> 01 Agosto 2016	
<b>S1: Sprint 1</b>	
<b>Requisito:</b> R1: Crear y editar cursos	
<b>Tareas de Desarrollo:</b>	
TD1. 1: Maqueta (wireframe) para crear cursos	
TD1. 2: Diagrama de clases de cursos	
TD1. 3: Programación de algoritmo para crear cursos	
TD1. 4: Algoritmo para registrar actividad de los estudiantes	

**Producto 23:** Ejemplo de Informe de las tareas de desarrollo de los requisitos del Sprint (P23).

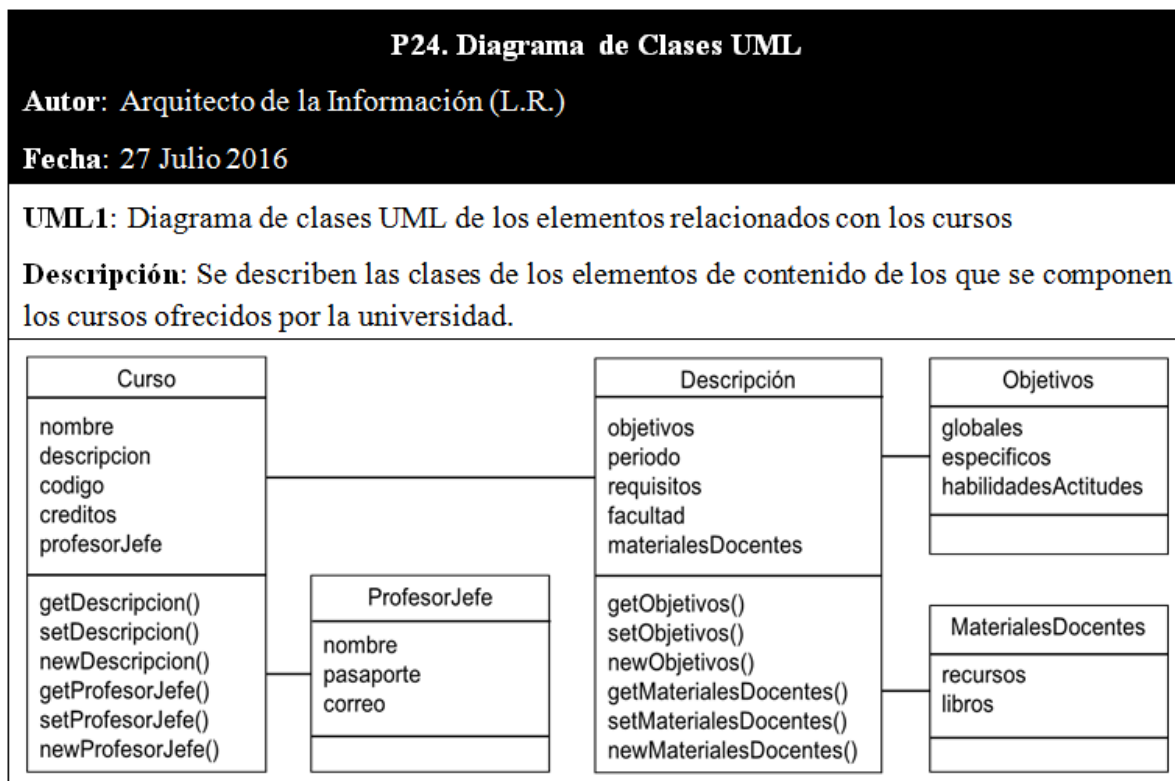


## P24. Diagrama de Clases UML

Este producto representa un diagrama, que identifica la estructura y el comportamiento de los objetos de contenido, identificados en el Modelo de contenido general del proyecto (P4), así como las relaciones entre estos objetos. Para cada diagrama de clases UML, se debe especificar los siguientes datos:

- **Identificador del diagrama de clases UML:** Adoptando la sigla **UMLx**, donde x corresponde al número correlativo y único, asignado para cada diagrama de clases UML del proyecto.
- **Nombre:** Nombre del diagrama de clases UML.
- **Descripción:** Descripción del diagrama de clases UML.
- **Diagrama de clases UML:** Representación visual de las relaciones entre las clases que involucran el Modelo de contenido general del proyecto (P4).

A continuación, se presenta un ejemplo concreto del producto “Diagrama de Clases UML”:



**Producto 24:** Ejemplo de Diagrama de Clases UML (P24).

**P25. Informe de las técnicas ágiles de la Arquitectura de la Información relacionadas con las tareas de desarrollo**

Este producto identifica las técnicas ágiles de la AI de las tareas de desarrollo. Para cada ítem, el producto debe especificar lo siguiente:

- **Identificador de la tarea de desarrollo:** Utilizando la sigla **TDx.y**, definida en el producto P23 (Informe de las tareas de desarrollo de los requisitos del Sprint).
- **Técnicas ágiles de la AI:** Listado de las técnicas ágiles de la AI, asignadas a la tarea de desarrollo.
- **Nombre:** Nombre de la tarea de desarrollo.

A continuación, se presenta un ejemplo concreto del producto “Informe de las técnicas ágiles de la Arquitectura de la Información relacionadas con las tareas de desarrollo”:

<b>P25. Informe de las técnicas ágiles de la Arquitectura de la Información relacionadas con las tareas de desarrollo</b>	
<b>Autor:</b> Equipo de Desarrollo (M.R. y A.G.)	
<b>Fecha:</b> 01 Agosto 2016	
<b>S1: Sprint 1</b>	
<b>Requisito:</b> R1: Crear y editar cursos	
<b>Tareas de Desarrollo</b>	<b>Técnica ágil de la AI</b>
TD1. 1: Maqueta (wireframe) para crear cursos	InterArch-T, Lo-fi prototyping
TD1. 2: Diagrama de clases de cursos	InterArch-T
<b>Requisito:</b> R2: Gestionar cursos en un entorno integrado	
<b>Tareas de Desarrollo</b>	<b>Técnica ágil de la AI</b>
TD2. 1: Inventario de contenido (Estudiante, Curso, Instructores)	Background investigation, Interviews, Meetings
TD2. 2: Vocabulario controlado	Consistency inspection, Participatory design

**Producto 25:** Ejemplo de Informe de las técnicas ágiles de la Arquitectura de la Información relacionadas con las tareas de desarrollo (P25).

### P26. Informe sobre la estimación de las tareas de desarrollo

Este producto identifica la duración estimada para cada tarea de desarrollo. Para cada ítem, el producto debe indicar lo siguiente:

- **Identificador de la tarea de desarrollo:** Utilizando la sigla **TDx.y**, definida en el producto P23 (Informe de las tareas de desarrollo de los requisitos del Sprint).
- **Nombre:** Nombre de la tarea de desarrollo.
- **Estimación:** Valoración numérica del tiempo requerido para implementar la tarea de desarrollo.
- **Tipo de estimación:** Tipo de estimación de la tarea de desarrollo. Ejemplo: Días, Horas o Minutos.

A continuación, se presenta un ejemplo concreto del producto “Informe sobre la estimación de las tareas de desarrollo”:

P26. Informe sobre la estimación de las tareas de desarrollo	
Autor: Equipo de Desarrollo (M.R. y A.G.)	
Fecha: 01 Agosto 2016	
S1: Sprint 1	
Requisito: R1: Crear y editar cursos	
Tareas de Desarrollo	Estimación
TD1. 1: Maqueta (wireframe) para crear cursos	8 Horas
TD1. 2: Diagrama de clases de cursos	4 Horas
TD1. 3: Programación de algoritmo para crear cursos	16 Horas
TD1. 4: Algoritmo para registrar actividad de los estudiantes	8 Horas
Requisito: R2: Gestionar cursos en un entorno integrado	
Tareas de Desarrollo	Estimación
TD2. 1: Inventario de contenido (Estudiante, Curso, Instructores)	8 Horas
TD2. 2: Vocabulario controlado	4 Horas
TD2. 3: Blueprint (entorno de enseñanza)	8 Horas

**Producto 26:** Ejemplo de Informe sobre la estimación de las tareas de desarrollo (P26).

### P27. Informe de los responsables de las tareas de desarrollo

Este producto identifica al responsable para llevar a cabo cada tarea. Para cada ítem, el producto debe especificar lo siguiente:

- **Identificador de la tarea de desarrollo:** Utilizando la sigla **TDx.y**, definida en el producto P23 (Informe de las tareas de desarrollo de los requisitos del Sprint).
- **Nombre:** Nombre de la tarea de desarrollo.
- **Identificador del miembro del Equipo de Desarrollo:** Utilizando la sigla **MEDx**, definida en el producto P19 (Informe sobre la disponibilidad del Equipo de Desarrollo para el Sprint).
- **Nombre responsable:** Nombre del miembro del Equipo de Desarrollo.

A continuación, se presenta un ejemplo concreto del producto “Informe de los responsables de las tareas de desarrollo”:

P27. Informe de los responsables de las tareas de desarrollo	
Autor: Equipo de Desarrollo (M.R. y A.G.)	
Fecha: 01 Agosto 2016	
S1: Sprint 1	
Requisito: R1: Crear y editar cursos	
Tareas de Desarrollo	Responsable
TD1. 1: Maqueta (wireframe) para crear cursos	Luis Rojas
TD1. 2: Diagrama de clases de cursos	Mario Reyes
TD1. 3: Programación de algoritmo para crear cursos	Adams Galkus
TD1. 4: Algoritmo para registrar actividad de los estudiantes	Mario Reyes
Requisito: R2: Gestionar cursos en un entorno integrado	
Tareas de Desarrollo	Responsable
TD2. 1: Inventario de contenido (Estudiante, Curso, Instructores)	Luis Rojas
TD2. 2: Vocabulario controlado	Luis Rojas
TD2. 3: Blueprint (entorno de enseñanza)	Luis Rojas
TD2. 4: Modelo conceptual de la base de datos	Adams Galkus
TD2. 5: Maqueta (wireframe) del entorno de aprendizaje	Luis Rojas

**Producto 27:** Ejemplo de Informe de los responsables de las tareas de desarrollo (P27).

### P28. Diagrama de la planificación diaria de las tareas de desarrollo

Diagrama que identifica el esfuerzo estimado de las tareas de desarrollo, detalladas en el “Informe de las tareas de desarrollo de los requisitos del Sprint” (P23), para cada día del Sprint. Mientras que en la primera columna del diagrama se listan las tareas de desarrollo identificadas para el Sprint, definidas en el producto P23, en la segunda columna del diagrama se establece la estimación de cada una de las tareas, identificadas en el producto P26. Finalmente, en la intersección de cada fila con cada columna, se identifica el esfuerzo (estimado en horas) de la tarea (fila) para un determinado día del Sprint (columna).

A continuación, se presenta un ejemplo concreto del producto “Diagrama de la planificación diaria de las tareas de desarrollo”:

P28. Diagrama de la planificación diaria de las tareas de desarrollo												
Autor: Equipo de Desarrollo (M.R. y A.G.)												
Fecha: 01 Agosto 2016												
S1: Sprint 1												
Tareas de Desarrollo	Est.	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	
Maqueta (wireframe) para crear cursos	8							8				
Diagrama de clases de cursos	4							4				
Programación de algoritmo para crear cursos	16								8	8		
Algoritmo para registrar actividad de los estudiantes	8								4	4		
Inventario de contenido (Estudiante, Curso, Instructores)	8	4	4									
Vocabulario controlado	4	4										
Blueprint (entorno de enseñanza)	8		6	2								
Modelo conceptual de la base de datos	6						6					
Maqueta (wireframe) del entorno de aprendizaje	12			4	8							
Modelo de contenido de estudiantes e instructores	6		6									
Diagrama de clases de estudiantes e instructores	4			4								
Maqueta (wireframe) para gestión de perfiles	8				4	4						
Programación de algoritmo para gestión de perfiles	12									2	10	
<b>Total</b>	<b>104</b>	<b>8</b>	<b>16</b>	<b>10</b>	<b>12</b>	<b>4</b>	<b>6</b>	<b>12</b>	<b>12</b>	<b>14</b>	<b>10</b>	

**Producto 28:** Ejemplo de Diagrama de la planificación diaria de las tareas de desarrollo (P28).

### P29. Informe de la revisión de las tareas de desarrollo

Este producto identifica las observaciones del estado actual de las tareas. Para cada ítem, el producto debe indicar lo siguiente:

- **Identificador del Sprint:** Utilizando el acrónimo **Sx**, definido en el producto P16 (Informe sobre la definición de “Terminado” para el Sprint).
- **Sprint:** Nombre del Sprint.
- **Día del Sprint:** Fecha del Sprint.
- **Identificador de la tarea de desarrollo:** Utilizando la sigla **TDx.y**, definida en el producto P23 (Informe de las tareas de desarrollo de los requisitos del Sprint).
- **Nombre:** Nombre de la tarea de desarrollo.
- **Observaciones:** Observaciones de la tarea de desarrollo.

A continuación, se presenta un ejemplo concreto del producto “Informe de la revisión de las tareas de desarrollo”:

<b>P29. Informe de la revisión de las tareas de desarrollo</b>	
<b>Autor:</b> Equipo de Desarrollo (M.R. y A.G.)	
<b>Fecha:</b> 10 Agosto 2016	
<b>S1:</b> Sprint 1	
<b>Día del Sprint:</b> Día 3 del Sprint (Miércoles 10 de agosto de 2016)	
<b>Tareas de Desarrollo</b>	<b>Observaciones</b>
TD1. 1: Maqueta (wireframe) para crear cursos	Tarea no comenzada. Planificada para el día 7 del Sprint. La maqueta debe soportar los elementos definidos en el diagrama de clases (TD1. 2).
TD1. 2: Diagrama de clases de cursos	Tarea no comenzada. Planificada para el día 6 del Sprint. Se debe utilizar, como base, el resultado del Producto de Validación 24.
TD1. 3: Programación de algoritmo para crear cursos	Tarea no comenzada. Planificada para los días 8 y 9 del Sprint.
TD1. 4: Algoritmo para registrar actividad de los estudiantes	Tarea no comenzada. Planificada para los días 8 y 9 del Sprint.

**Producto 29:** Ejemplo de Informe de la revisión de las tareas de desarrollo (P29).

### **P30. Informe sobre la revisión del incremento del producto**

Este producto describe las observaciones realizadas en la revisión del incremento. Para cada Sprint, el producto debe indicar lo siguiente:

- **Identificador del Sprint:** Utilizando el acrónimo **Sx**, definido en el producto P16 (Informe sobre la definición de “Terminado” para el Sprint).
- **Sprint:** Nombre del Sprint.
- **Observaciones:** Descripción de las observaciones del incremento.

A continuación, se presenta un ejemplo concreto del producto “Informe sobre la revisión del incremento del producto”:

<b>P30. Informe sobre la revisión del incremento del producto</b>	
<b>Autor:</b> Equipo de Desarrollo (M.R. y A.G.)	
<b>Fecha:</b> 22 Agosto 2016	
S1: Sprint 1	
<p><b>Observaciones:</b> Un estudiante de la universidad ha participado para revisar el diseño del entorno Web que permite gestionar y configurar los cursos. El estudiante se ha manifestado conforme con la composición de los elementos estructurales del contenido, pero ha indicado la necesidad de que el SI tenga una cuenta de usuario, integrada con los demás servicios ofrecidos por la universidad (autenticación en los ordenadores, VPN, correo electrónico y biblioteca).</p> <p>Asimismo, un trabajador de la universidad (del área administrativa) ha participado para validar las funcionalidades que permiten configurar las preferencias de los estudiantes e instructores. En este caso, el personal administrativo ha señalado que los datos recopilados permiten identificar las prioridades más relevantes de los estudiantes y profesores. No obstante, se han hecho sugerencias para que estas funcionalidades se extiendan a los estudiantes, con el fin de que éstos puedan autoconfigurar sus preferencias.</p>	

**Producto 30:** Ejemplo de Informe sobre la revisión del incremento del producto (P30).

### **P31. Informe sobre la inspección de las prácticas de trabajo**

Este producto describe las mejoras que se podrían introducir en el siguiente Sprint. Para cada Sprint, el producto debe especificar lo siguiente:

- **Identificador del Sprint:** Utilizando el acrónimo **Sx**, definido en el producto P16 (Informe sobre la definición de “Terminado” para el Sprint).
- **Sprint:** Nombre del Sprint.
- **Observaciones:** Descripción de las mejoras para considerar en el siguiente Sprint.

A continuación, se presenta un ejemplo concreto del producto “Informe sobre la inspección de las prácticas de trabajo”:

<b>P31. Informe sobre la inspección de las prácticas de trabajo</b>	
<b>Autor:</b> Equipo de Desarrollo (M.R. y A.G.)	
<b>Fecha:</b> 22 Agosto 2016	
S1: Sprint 1	
<b>Observaciones:</b> El Equipo de Desarrollo ha manifestado la necesidad de incorporar elementos cualitativos en los modelos de contenido, elaborados por el Arquitecto de la Información, con el fin de que se incorpore información sobre los criterios de aceptación de los usuarios finales, para que los miembros del Equipo de Desarrollo puedan analizarla.	

**Producto 31:** Ejemplo de Informe sobre la inspección de las prácticas de trabajo (P31).



### 3.3. QMPSR

Como se ha descrito en la sección anterior, en el marco de integración de Scrum-UIA, se prescriben diferentes actividades, tareas y productos que permiten integrar la Arquitectura de la Información dentro de un proceso ágil de desarrollo centrado en el usuario. Asimismo, se proporciona un panorama general del proyecto que facilita obtener una visión integradora del producto, con respecto a las prioridades de la usabilidad, de la AI y del valor de negocio.

No obstante, se debe asegurar que la ejecución de los requisitos en Scrum-UIA responda a las prioridades definidas en el panorama general del proyecto, con el fin de que el objetivo general se complete concentrándose en los requisitos más importantes. Por lo tanto, en esta sección se presenta en detalle el método de priorización propuesto para la metodología Scrum-UIA, que consigue adaptarse a entornos cambiantes y concentrarse en los requisitos más importantes para integrar la Arquitectura de la Información en un marco ágil de desarrollo centrado en el usuario, lo que también permitirá corroborar la hipótesis de partida **H2.2**: Es posible idear un método de priorización que permita a la metodología anteriormente ideada (hipótesis de partida **H2.1**) conducir las prioridades de los requisitos a través de las preferencias de la usabilidad, de la Arquitectura de la Información y del valor de negocio, generando una clasificación final que permita reducir las colisiones de requisitos y presentar un buen comportamiento en términos de escalabilidad.

A continuación, se describe formalmente el método de priorización propuesto para Scrum-UIA.

#### 3.3.1. Descripción Formal de QMPSR

Con el fin de hacer frente a la exigencia de conducir adecuadamente la implementación de los requisitos de la metodología Scrum-UIA y resolver las limitaciones de los métodos de priorización actuales (identificadas en la sección 2.5), se propone un método cualitativo para la priorización de requisitos de software, llamado QMPSR (**Q**ualitative **M**ethod for **P**rioritizing **S**oftware **R**equirements). Este método permite formalizar la evaluación de la priorización de los requisitos y centrar el proceso de priorización de Scrum-UIA a través de las prioridades relacionadas con la usabilidad, la AI y el valor de negocio.

En la Figura 17, se presenta QMPSR. Como se puede observar, este método incluye cuatro fases principales que tienen como objetivo dirigir el proceso de priorización de los requisitos de Scrum-UIA (realizado en la Tarea 2.2.4), considerando las prioridades establecidas en el panorama general (identificadas en la Tarea 1.4.1). De esta forma, los aspectos relevantes del proyecto se componen por un conjunto de elementos, y la priorización de los requisitos se materializa de acuerdo con los elementos de los aspectos relevantes del proyecto. A continuación, se describen en detalle las diferentes fases de QMPSR.

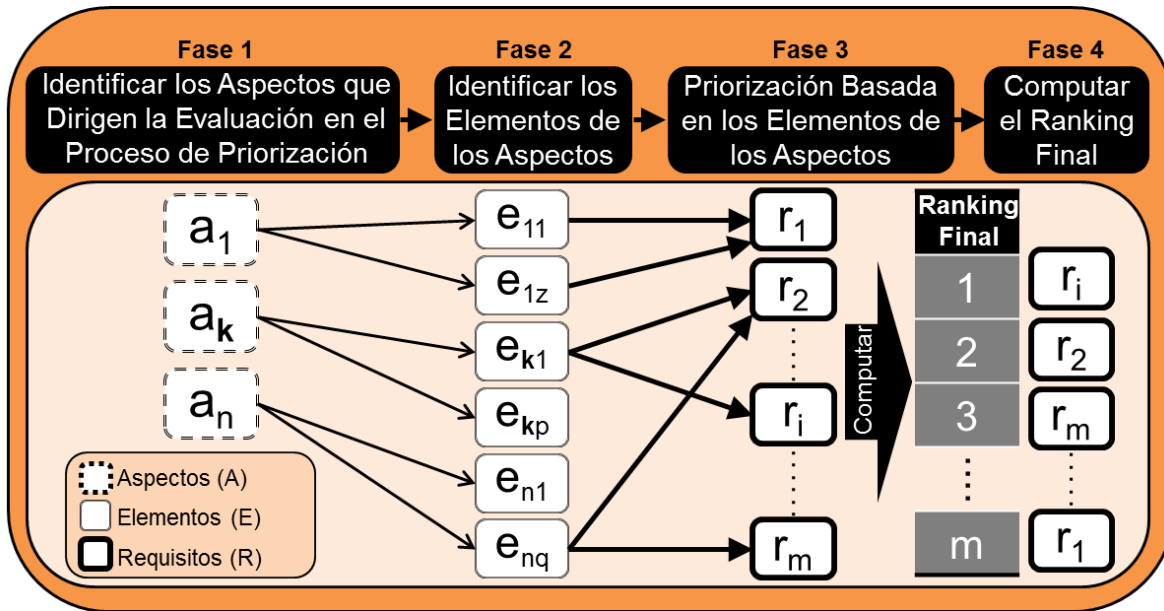


Figura 17: Fases de QMPSR.

### 3.3.1.1. Fase 1: Identificar los Aspectos Relevantes del Proyecto que Conducen la Evaluación en el Proceso de Priorización de los Requisitos

La primera fase del método QMPSR tiene como objetivo identificar los aspectos relevantes del proyecto, y se compone de tres pasos:

#### a) Identificar y definir los aspectos relevantes del proyecto

El primer paso consiste en determinar los aspectos relevantes del proyecto. Estos aspectos corresponden a los asuntos más importantes de los proyectos que conducen la evaluación en la priorización de los requisitos. En otras palabras, éstas son las características que las personas con capacidad decisoria (participantes del proceso de priorización de Scrum-UIA) consideran a la hora de priorizar los requisitos que son valiosos para el proyecto. Por lo tanto, el primer paso tiene como objetivo determinar formalmente los aspectos más importantes del proyecto.

En QMPSR, estos aspectos se indican de forma genérica, y son negociados colectivamente entre el Arquitecto de la Información y el Dueño del Producto en la Tarea 1.4.1 del Grupo de Actividades de la Investigación Contextual de Scrum-UIA (ver sección 3.2.3.1). Más específicamente, QMPSR permite hacer frente a un número ilimitado de aspectos, que se pueden definir para un proyecto concreto bajo demanda.

Cabe destacar que Riegel y Doerr (Riegel y Doerr, 2015) identificaron, por medio de una revisión sistemática de la bibliografía, qué aspectos (criterios de priorización) se deben tomar en consideración para determinar el valor de los requisitos. Por lo tanto, el modelo propuesto por (Riegel y Doerr, 2015) permite apoyar la identificación y definición de los

aspectos relevantes del proyecto, que se deben determinar en este paso. Por ejemplo, de acuerdo con el modelo propuesto por (Riegel y Doerr, 2015), con el fin de discernir el nivel de relevancia de los requisitos, los proyectos software deben considerar principalmente, pero no exclusivamente, aspectos tales como *la Usabilidad, el Beneficio Relacionado con la Estrategia y el Valor de Negocio*.

**b) Definir la prioridad de los aspectos relevantes del proyecto**

El segundo paso de esta primera fase del método QMPSR consiste en definir la prioridad de los aspectos relevantes del proyecto. Esta prioridad se establece en términos del conjunto de los aspectos identificados en el proyecto, generando un ranking de ellos, ordenados por relevancia. De este modo, las prioridades se asignan colectivamente entre los tomadores de decisiones (definidas en la Tarea 1.4.1 del Grupo de Actividades de la Investigación Contextual de Scrum-UIA), con el fin de identificar las preferencias. Por ejemplo, para un total de  $n$  aspectos relevantes, definidos en un proyecto, el aspecto con la relevancia más baja es asignado con un 1, mientras que el aspecto de mayor relevancia es asignado con  $n$ .

**c) Definir la Prioridad Normalizada de los Aspectos Relevantes del Proyecto**

El último paso de esta fase consiste en definir la Prioridad Normalizada de los Aspectos Relevantes del Proyecto ( $P$ ), que se utiliza para representar la prioridad con respecto al número total de aspectos relevantes definidos en el proyecto. Por lo tanto, esta definición normalizada de la prioridad de los aspectos se adopta para garantizar que  $P$  tenga valores entre 0 y 1.

Por ejemplo, sea  $A = \{a_1, \dots, a_k, \dots, a_n\}$  una colección finita de los aspectos relevantes de un proyecto, donde  $a_k \in A$ , tal que  $k$  representa el orden de prioridad de importancia y  $n$  es el número total de los aspectos relevantes del proyecto definidos  $\{n \in \mathbb{N}: n \geq 1\}$ . La Prioridad Normalizada de un Aspecto Relevante del Proyecto  $a_k$  se define formalmente como:

$$P(a_k) = \frac{k}{|A|}, \quad (1)$$

donde  $P(a_h) > P(a_k)$  significa que el aspecto  $a_h$  ( $a_h \in A$ ) tiene mayor prioridad que  $a_k$ .

**3.3.1.2. Fase 2: Identificar los Elementos de los Aspectos Relevantes del Proyecto**

La segunda fase del QMPSR tiene como objetivo configurar los elementos relacionados con los aspectos relevantes del proyecto (identificados en la fase anterior). Esta fase se compone de dos pasos:

**a) Identificar y definir los elementos de aspectos relevantes del proyecto**

Cada aspecto relevante del proyecto incluye un conjunto de elementos comunes, que se utilizan para conducir la valoración de la priorización de requisitos. Por lo tanto, el primer paso de la segunda fase de QMPSR consiste en identificar y describir los elementos para cada aspecto definido en fase anterior.

Estos elementos se negocian colectivamente entre los tomadores de decisiones (Arquitecto de la Información y Dueño del Producto), y se determinan en la Tarea 1.4.1 del Grupo de Actividades de la Investigación Contextual de Scrum-UIA (ver sección 3.2.3.1). Cabe resaltar también, que los aspectos identificados por (Riegel y Doerr, 2015) se pueden refinar y dividir en diferentes subcategorías, lo que permite apoyar la identificación y definición de elementos específicos para los aspectos relevantes del proyecto. Por ejemplo, de acuerdo con el modelo propuesto por (Riegel y Doerr, 2015), en un proyecto software de comercio electrónico, algunos de los elementos relacionados con el aspecto relevante del *Valor de Negocio* pueden corresponder a: *Servicio de Gestión de Ventas*, *Gestión de Stocks* y *Flujos de Ingresos Adicionales*, entre otros. Asimismo, QMPSR permite hacer frente a un número ilimitado de elementos, que se pueden indicar de forma genérica, y asociar a los aspectos definidos para un proyecto concreto.

**b) Definir la prioridad de los elementos asignados a cada uno de los aspectos relevantes del proyecto**

Este paso consiste en identificar la prioridad de los elementos asignados a cada aspecto definido anteriormente. Para ello, se consideran tres prioridades diferentes: *alta*, *media* o *baja*. Esto permite comparar la prioridad entre los distintos elementos de aspecto, independientemente de su número.

Por ejemplo, sea  $E = \{e_{11}, \dots, e_{1z}, \dots, e_{k1}, \dots, e_{kp}, \dots, e_{n1}, \dots, e_{nq}\}$  una colección finita de elementos, donde  $z$ ,  $p$  y  $q$  corresponden al número total de elementos para los aspectos  $a_1$ ,  $a_k$  y  $a_n$ , respectivamente,  $\{z, p, q \in \mathbb{N}: z, p, q \geq 1\}$ . La prioridad del elemento  $e_{kp}$  se define formalmente en términos de la función  $L(e_{kp})$ , donde  $L: E \rightarrow \{1, 2, 3\}$ , con la siguiente interpretación de sus valores:

$$L(e_{kp}) = \begin{cases} 1, & \text{si el elemento } e_{kp} \text{ tiene una prioridad } \textit{baja}; \\ 2, & \text{si el elemento } e_{kp} \text{ tiene una prioridad } \textit{media}; \\ 3, & \text{si el elemento } e_{kp} \text{ tiene una prioridad } \textit{alta}. \end{cases} \quad (2)$$

**3.3.1.3. Fase 3: Proceso de Priorización de los Requisitos Basado en los Elementos de los Aspectos Relevantes del Proyecto**

La tercera fase en QMPSR tiene como objetivo generar la priorización de los requisitos a través de los elementos de los aspectos relevantes del proyecto. Por lo tanto, la priorización

de los requisitos se realiza mediante la identificación de las relaciones entre los elementos de los aspectos relevantes del proyecto y los requisitos. En este caso, un elemento puede estar asociado a uno o varios requisitos diferentes, y un requisito puede estar asociado a uno, muchos o ningún elemento de los aspectos relevantes del proyecto. Por ejemplo, un requisito relacionado con la gestión de los activos de la empresa, que están listos para la venta, puede estar asociado con el elemento de *Gestión del Inventario* del aspecto *Valor de Negocio*.

Tales relaciones se especifican y negocian colectivamente entre el Dueño del Producto y el Arquitecto de la Información durante la Tarea 2.2.4 (ver sección 3.2.3.2). Concretamente, el Arquitecto de la Información tiene como objetivo priorizar los requisitos (relacionar los elementos de los aspectos relevantes del proyecto y los requisitos) para dar una mayor prioridad a los que evolucionan hacia el dominio de la solución y facilitan el desarrollo incremental en Scrum-UIA, dirigido por la AI (en la sección 3.4 se describe una técnica que permite fomentar el desarrollo incremental de Scrum-UIA).

A continuación, se describe formalmente la relación entre los requisitos y los elementos de los aspectos relevantes del proyecto:

Sea  $R = \{r_1, \dots, r_i, \dots, r_m\}$  una colección finita de requisitos, donde  $r_i$  es un requisito del proyecto y  $m$  es el número total de los requisitos definidos  $\{m \in \mathbb{N}: m \geq 1\}$ . La relación entre el requisito  $r_i$  y el elemento  $e_{kv}$  ( $e_{kv} \in E$ ) se define formalmente en términos de la función  $C(e_{kv}, r_i)$ , donde  $C: ExR \rightarrow \{0,1\}$ , con la siguiente interpretación para cada uno de estos valores:

$$C(e_{kv}, r_i) = \begin{cases} 1, & \text{si el elemento } e_{kv} \text{ está relacionado con el requisito } r_i; \\ 0, & \text{en caso contrario.} \end{cases} \quad (3)$$

#### 3.3.1.4. Fase 4: Computar la Clasificación Final

La última fase en QMPSR consiste en generar una clasificación final de los requisitos, basada en los aspectos relevantes del proyecto. Esta fase incluye dos pasos:

##### a) Computar el nivel de relevancia de los requisitos por aspecto

Este paso consiste en determinar el nivel de relevancia de los requisitos, de acuerdo con los aspectos del proyecto. El nivel de relevancia de un requisito por aspecto se expresa en valor porcentual, y es agrupado por aspecto. Más específicamente, la relevancia de un requisito particular, con respecto a un aspecto específico, corresponde al porcentaje que representa la suma de las prioridades de los elementos del aspecto relevante del proyecto, asociado con el requisito más un *Factor de Asociación* ( $G$ ). De esta manera,  $G$  corresponde a la relación entre el número de los requisitos relacionados con el elemento y el número máximo de requisitos con el que un elemento del mismo aspecto relevante puede estar asociado. Por lo tanto,  $G$  permite diferenciar el nivel de relevancia entre los elementos con la misma

prioridad en función del número de requisitos asociados. En consecuencia, los elementos con la misma prioridad, pero con diferente número de requisitos asociados tienen diferente nivel de relevancia.

Así pues, el número total de los requisitos relacionados con el elemento  $e_{kv}$  se define como:

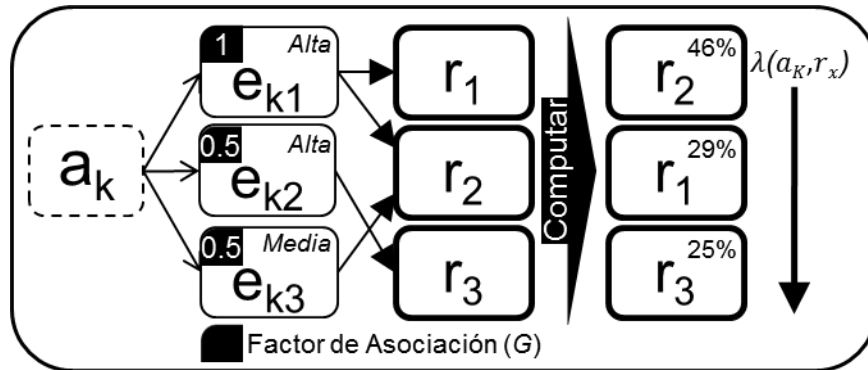
$$TC(e_{kv}) = \sum_{i=1}^m C(e_{kv}, r_i), \quad (4)$$

donde  $C(e_{kv}, r_i)$  identifica si el elemento  $e_{kv}$  está relacionado con el requisito  $r_i$ , tal y como se define en la fórmula (3), y  $m$  indica el número total de requisitos definidos. Por lo tanto, el Factor de Asociación del elemento  $e_{kv}$  se define como:

$$G(e_{kv}, e_{kb}) = \frac{TC(e_{kv})}{TC(e_{kb})}, \quad (5)$$

donde  $TC(e_{kb})$  es el número máximo de requisitos con el que un elemento del aspecto  $a_k$  se ha asociado, tal que  $e_{kb} \in E, \forall e_{kx} \in E, TC(e_{kb}) \geq TC(e_{kx})$ .

En la Figura 18, se ilustra la priorización de requisitos, donde se identifica la implicación de  $G$  en el *Nivel de Relevancia de los Requisitos por Aspecto* ( $\lambda$ ). Como se puede observar en la Figura 18, los requisitos  $r_1$  y  $r_3$  están asociados con los elementos  $e_{k1}$  y  $e_{k2}$ , respectivamente, que tienen la misma prioridad (*alta*), pero diferente  $\lambda$ , debido a que el  $G$  es diferente para cada elemento (1 y 0,5, respectivamente). Por lo tanto, un mayor número de requisitos asociados con un elemento corresponde a una mayor  $G$  y, consecuentemente, su prioridad es más alta. También se puede observar que el requisito  $r_2$  obtiene el mayor  $\lambda$  en  $a_k$ . Este nivel de relevancia es justificado por la asociación de elementos  $e_{k1}$  y  $e_{k3}$  con el requisito  $r_2$ .



**Figura 18:** La influencia del Factor de Asociación ( $G$ ) sobre el Nivel de Relevancia del Requisito por Aspecto ( $\lambda$ ).

Continuando con la descripción del nivel de relevancia de los requisitos por aspecto, la prioridad de un elemento  $e_{kv}$  del aspecto  $a_k$ , asociado con el requisito  $r_i$ , y teniendo también en cuenta su  $G$ , se define formalmente en términos de la función  $I: ExR \rightarrow \mathbb{R}$ , que se representa de la siguiente forma:

$$I(e_{kv}, r_i) = \begin{cases} L(e_{kv}) + G(e_{kv}, e_{kb}), & \text{si } C(e_{kv}, r_i) = 1; \\ 0, & \text{en caso contrario.} \end{cases} \quad (6)$$

Aquí  $L(e_{kv})$  define la función de prioridad para el elemento  $e_{kv}$ , según la fórmula (2), mientras que  $G(e_{kv}, e_{kb})$  define el factor de asociación para el elemento  $e_{kv}$ , computado con la fórmula (5). Además,  $C(e_{kv}, r_i)$  identifica si el elemento  $e_{kv}$  está relacionado con el requisito  $r_i$ , como se define en la fórmula (3).

De este modo, se puede obtener los elementos asociados a los requisitos. Por ejemplo, sea  $E_k = \{e_{k1}, \dots, e_{kp}\}$  una subcolección finita de todos los elementos relacionados con aspecto  $a_k$  ( $E_k \subset E$ ), donde  $|E_k|$  corresponde al número total de ellos ( $p = |E_k|$ ). Por lo tanto, el número total de elementos del aspecto  $a_k$ , asignado al requisito  $r_i$ , se define como:

$$TI(a_k, r_i) = \sum_{v=1}^{|E_k|} I(e_{kv}, r_i), \quad (7)$$

donde  $|E_k|$  indica el número total de los elementos definidos por el aspecto  $a_k$ . Finalmente, el nivel de relevancia del requisito  $r_i$ , con respecto al aspecto  $a_k$ , en términos porcentuales se define formalmente como:

$$\lambda(a_k, r_i) = \frac{TI(a_k, r_i)}{\sum_{v=1}^{|E_k|} ((L(e_{kv}) + G(e_{kv}, e_{kb}))TC(e_{kv}))} 100, \quad (8)$$

donde  $\lambda(a_k, r_i) > \lambda(a_k, r_j)$  significa que el requisito  $r_i$  tiene mayor prioridad que el requisito  $r_j$  para el aspecto  $a_k$ . Por ejemplo, en la Figura 18, se puede observar que el nivel de relevancia de los requisitos  $r_1$ ,  $r_2$  y  $r_3$ , con respecto al aspecto  $a_k$ , obtuvo 29%, 46% y 25%, respectivamente, como resultado de la aplicación de  $\lambda$ .

**b) Calcular la clasificación final de los requisitos de acuerdo con el nivel de relevancia de cada requisito por aspecto**

El último paso consiste en generar la *Clasificación Final de Requisitos (FR)*, teniendo en cuenta su nivel de relevancia en todos los aspectos definidos en el proyecto. *FR* se calcula a través de dos etapas sucesivas: la primera etapa tiene como objetivo generar la *Clasificación de Relevancia de los Requisitos por Aspecto (W)*, y el segundo paso trata de ponderar  $W$  a través de  $P$ .

Así pues, para obtener  $W$ , es necesario calcular una lista ordenada de los requisitos, de acuerdo con  $\lambda$ . De esta manera, el orden de relevancia entre los requisitos  $r_i$  y  $r_j$ , con respecto al aspecto  $a_k$ , se define formalmente en términos de la función  $M\lambda(a_k, r_i, r_j)$ , donde  $M\lambda: AxRxR \rightarrow \{0,1\}$ , con la siguiente interpretación de los valores:

$$M\lambda(a_k, r_i, r_j) = \begin{cases} 1, & \text{si } \lambda(a_k, r_i) > \lambda(a_k, r_j); \\ 0, & \text{en caso contrario.} \end{cases} \quad (9)$$

Donde  $\lambda(a_k, r_i)$  define la función de nivel de relevancia para el requisito  $r_i$  con respecto al aspecto  $a_k$  en términos porcentuales, tal y como se define en la fórmula (8). De esta manera, el número total de los requisitos que tienen un nivel de relevancia más alto que el requisito  $r_i$ , con respecto al aspecto  $a_k$ , se define como:

$$TM\lambda(a_k, r_i) = \sum_{j=1}^m M\lambda(a_k, r_i, r_j), \quad (10)$$

donde  $M\lambda(a_k, r_i, r_j)$  define el orden de relevancia para los requisitos  $r_i$  y  $r_j$  con respecto al aspecto  $a_k$ , como se describe en la fórmula (9), y  $m$  representa el número total de los requisitos. Por lo tanto, la clasificación del requisito  $r_i$  para el aspecto  $a_k$  se define formalmente en términos de la función  $W$ , donde  $W: AxR \rightarrow \{1, \dots, m\}$  y  $\{W(a_k, r_i) \in \mathbb{N}; 1 \leq W(a_k, r_i) \leq m\}$ , de modo que  $m$  es el número total de los requisitos del proyecto. Esta clasificación se lleva a cabo mediante la aplicación de un conjunto de reglas, conducidas por el *Nivel de Relevancia de los Requisitos por Aspecto* ( $\lambda$ ):

$$W(a_k, r_i) = \begin{cases} 1, & \text{si } \lambda(a_k, r_i) = 0; \\ m - TM\lambda(a_k, r_i), & \text{en caso contrario.} \end{cases} \quad (11)$$

$\lambda(a_k, r_i)$  define el nivel de relevancia para el requisito  $r_i$ , con respecto al aspecto relevante  $a_k$ , en términos porcentuales, tal y como se define en la fórmula (8). Por otra parte,  $TM\lambda(a_k, r_i)$  identifica el número total de los requisitos, que tienen un nivel de relevancia más alto que el requisito  $r_i$ , con respecto al aspecto  $a_k$ , computado con la fórmula mostrada en (10).

Por último, la clasificación final del requisito  $r_i$ , tomando en consideración todos los aspectos relevantes del proyecto, se define formalmente como:

$$FR(r_i) = \sum_{h=1}^n W(a_h, r_i)P(a_h), \quad (12)$$



donde  $P(a_h)$  define la función de Prioridad Normalizada para el Aspecto Relevante del Proyecto  $a_h$ , de acuerdo con la fórmula (1), y  $n$  es el número total de aspectos relevantes definidos en el proyecto. Finalmente,  $FR(r_i) > FR(r_j)$  significa que el requisito  $r_i$  tiene una mayor prioridad en la clasificación final que el requisito  $r_j$ . Sin embargo,  $FR(r_i) = FR(r_j)$  significa que hay una colisión entre los requisitos  $r_i$  y  $r_j$ , que se abordará en detalle en la sección 5.1.

### 3.3.2. Ejemplo de Aplicación

Una vez descrito formalmente el método de priorización propuesto para Scrum-UIA, en esta sección se detalla un ejemplo de aplicación, con el objetivo de describir más en profundidad el funcionamiento de QMPSR.

#### a) Descripción de la tarea para el ejemplo de aplicación

Se trata de un ejemplo de aplicación, en el que el Arquitecto de la Información y el Dueño del Producto deben gestionar y priorizar un conjunto de requisitos de un Sistema de Información (SI) de gestión de aprendizaje de una universidad. El objetivo de este SI es que los estudiantes y los instructores (profesores) puedan gestionar sus cursos en un entorno Web, y que además los administradores de la universidad puedan gestionar la inscripción y la matrícula de los cursos. El conjunto de requisitos de este SI se puede apreciar en la Tabla 19. A partir de los requisitos proporcionados, el Arquitecto de la Información y el Dueño del Producto trabajarán en la priorización de los requisitos, llevando a cabo las siguientes tareas:

- *Identificación de los aspectos relevantes del proyecto* (Fase 1 de QMPSR).
- *Identificación de los elementos de los aspectos relevantes del proyecto* (Fase 2 de QMPSR).
- *Priorización de los requisitos en base a los elementos de los aspectos* (Fase 3 de QMPSR).
- *Generación del ranking final de los requisitos* (Fase 4 de QMPSR).

Requisitos del Proyecto
<b>R1:</b> Crear y editar cursos.  <b>Descripción:</b> Crear y editar los cursos ofrecidos por la universidad, con el fin de configurar sus datos generales (por ejemplo, nombre, objetivos, requisitos, profesores).  <b>Usuarios finales:</b> Instructores, Administradores.

Requisitos del Proyecto
<p><b>Análisis:</b> Este requisito permite definir y configurar las materias que forman los planes de estudio de la universidad.</p>
<p><b>R2:</b> Gestionar cursos en un entorno integrado.</p> <p><b>Descripción:</b> Proveer un entorno virtual que permite consultar y gestionar los diferentes cursos que se llevan a cabo.</p> <p><b>Usuarios finales:</b> Estudiantes.</p> <p><b>Análisis:</b> Este requisito permite que la información de aprendizaje de los estudiantes se pueda organizar y configurar.</p>
<p><b>R3:</b> Crear y editar actividades de evaluación.</p> <p><b>Descripción:</b> Generar actividades de evaluación, asociadas a los cursos impartidos por la universidad, con el fin de permitir su gestión electrónica en todas sus fases.</p> <p><b>Usuarios finales:</b> Instructores.</p> <p><b>Análisis:</b> Este requisito permite digitalizar las instancias de evaluación.</p>
<p><b>R4:</b> Configurar perfiles de estudiantes y profesores.</p> <p><b>Descripción:</b> Configurar datos personales y preferencias de los estudiantes y profesores.</p> <p><b>Usuarios finales:</b> Administradores.</p> <p><b>Análisis:</b> Este requisito permite digitalizar los perfiles de los estudiantes y profesores, y enlazarlos con otros entornos digitales.</p>
<p><b>R5:</b> Gestionar las inscripciones en los cursos.</p> <p><b>Descripción:</b> Proveer un entorno integrado que permite conducir el proceso de matrícula de los estudiantes.</p> <p><b>Usuarios finales:</b> Administradores.</p> <p><b>Análisis:</b> Este requisito facilita que la composición de los cursos se pueda tramitar.</p>
<p><b>R6:</b> Incorporar material en los cursos.</p> <p><b>Descripción:</b> Enlazar y proveer documentos en diferentes formatos, asociados a los cursos impartidos por la universidad.</p> <p><b>Usuarios finales:</b> Instructores.</p> <p><b>Análisis:</b> Este requisito permite definir los contenidos de los cursos.</p>

Requisitos del Proyecto
<p><b>R7:</b> Consultar los eventos y tareas de los cursos.</p> <p><b>Descripción:</b> Proveer una estructura que permita consultar y navegar a través de los eventos y las tareas de los cursos.</p> <p><b>Usuarios finales:</b> Estudiantes, Observadores.</p> <p><b>Análisis:</b> Este requisito provee información de los cursos a los estudiantes.</p>
<p><b>R8:</b> Consultar información sobre el progreso de los estudiantes.</p> <p><b>Descripción:</b> Generar informes sobre las interacciones de los estudiantes en los cursos y las estadísticas de uso del SI.</p> <p><b>Usuarios finales:</b> Instructores, Observadores.</p> <p><b>Análisis:</b> Este requisito permite monitorear el proceso de aprendizaje de los estudiantes.</p>
<p><b>R9:</b> Consultar evaluaciones de las actividades.</p> <p><b>Descripción:</b> Revisar los resultados de las actividades de evaluación y notificar sobre los ciclos requeridos.</p> <p><b>Usuarios finales:</b> Estudiantes, Observadores.</p> <p><b>Análisis:</b> Este requisito provee retroalimentación a los estudiantes respecto a los trabajos de evaluación realizados.</p>
<p><b>R10:</b> Enviar, de manera electrónica, las actividades de evaluación.</p> <p><b>Descripción:</b> Permitir adjuntar y relacionar documentos digitales con las actividades de evaluación.</p> <p><b>Usuarios finales:</b> Estudiantes.</p> <p><b>Análisis:</b> Este requisito facilita acceder a los trabajos de evaluación de los estudiantes y proporciona un registro automático del proceso.</p>
<p><b>R11:</b> Crear entornos de discusión.</p> <p><b>Descripción:</b> Proveer Chat y grupo de discusión entre los integrantes de los cursos.</p> <p><b>Usuarios finales:</b> Instructores.</p> <p><b>Análisis:</b> Este requisito permite promover la interacción y el trabajo en equipo de los estudiantes.</p>
<p><b>R12:</b> Proveer un repositorio de archivos.</p>

Requisitos del Proyecto
<p><b>Descripción:</b> Proveer un espacio virtual centralizado para almacenar y mantener información digital.</p> <p><b>Usuarios finales:</b> Estudiantes.</p> <p><b>Análisis:</b> Este requisito provee una capacidad de almacenamiento virtual para que los estudiantes puedan centralizar y gestionar los recursos.</p>
<p><b>R13:</b> Crear aulas virtuales.</p> <p><b>Descripción:</b> Proveer entorno virtual que permita realizar enseñanza en línea.</p> <p><b>Usuarios finales:</b> Instructores.</p> <p><b>Análisis:</b> Este requisito permite a los estudiantes acceder al material de estudio y, al mismo tiempo, interactuar con el profesor y con otros estudiantes.</p>
<p><b>R14:</b> Revisar el SI como otro usuario.</p> <p><b>Descripción:</b> Permitir simular el SI, de acuerdo a diferentes perfiles.</p> <p><b>Usuarios finales:</b> Administrador.</p> <p><b>Análisis:</b> Este requisito permite verificar las funcionalidades del SI.</p>

**Tabla 19:** Requisitos del SI de gestión de aprendizaje de la universidad.

La tarea inicial consiste en la *identificación de los aspectos relevantes del proyecto* (correspondiente a la Tarea 1.4.1 del Grupo de Actividades de la Investigación Contextual de Scrum-UIA). En esta tarea se deben definir los distintos aspectos relevantes del proyecto, generar un ranking de ellos ordenados por relevancia, y calcular la prioridad normalizada de cada aspecto relevante (ver sección 3.3.1.1). En la Tabla 20, se presenta el resultado de esta tarea. Como se puede observar, se han definido los aspectos *Usabilidad*, *Contenido* y *Valor del Negocio* para considerar de manera formal, durante el proceso de priorización, las prioridades relacionadas con los usuarios finales, la AI y el proyecto, respectivamente. Dichos aspectos se seleccionaron y priorizaron porque el tipo de aplicación del dominio (entorno educativo) hace que éstos sean más relevantes que, por ejemplo, el coste de implementación del proyecto, el riesgo de llevarlo a cabo o el beneficio económico esperado. Como se mencionó anteriormente, Riegel y Doerr (Riegel y Doerr, 2015) proveen un conjunto de aspectos que se pueden considerar para apoyar esta tarea. En Scrum-UIA, los aspectos se obtienen en base al análisis de las necesidades y de las preferencias del contexto, del contenido y de los usuarios finales del proyecto, adquiridos desde las actividades de la Investigación Contextual (específicamente, en las actividades 1.1, 1.2 y 1.3, respectivamente).

Por otro lado, los aspectos *Usabilidad*, *Contenido* y *Valor del Negocio* tienen una prioridad de 3, 2 y 1, respectivamente, donde 3 y 1 son las prioridades con mayor y menor importancia, respectivamente. Por lo tanto, estos aspectos tienen una prioridad normalizada (*P*) de 1, 0,66 y 0,33, respectivamente, tal y como se define en la fórmula (1). De este modo, *Usabilidad* y *Valor de Negocio* son los aspectos con mayor y menor prioridad, respectivamente. De este modo, el proceso de priorización de los requisitos se conduciría principalmente por las preferencias relacionadas con los usuarios finales del proyecto.

Aspecto	Prioridad	Prioridad Normalizada ( <i>P</i> )
Usabilidad	3	1
Contenido	2	0,66
Valor de Negocio	1	0,33

**Tabla 20:** Aspectos relevantes del SI de gestión de aprendizaje de la universidad.

La siguiente tarea corresponde a la *Identificación de los elementos de los aspectos relevantes del proyecto* (que pertenece a la Tarea 1.4.1 del Grupo de Actividades de la Investigación Contextual de Scrum-UIA). Esta tarea consiste en identificar y priorizar los elementos relacionados con los aspectos identificados en la tarea anterior (ver sección 3.3.1.2). En la Tabla 21, se presenta el resultado de los elementos identificados para cada aspecto relevante del proyecto. De la misma forma que se obtuvieron los aspectos, estos elementos también se han seleccionado y priorizado porque son más relevantes para el tipo de aplicación del dominio (entorno educativo). Por ejemplo, para el aspecto *Usabilidad* se definió el elemento *Proveer facilidad de aprendizaje*, con el objetivo de considerar formalmente la importancia de proveer una interacción efectiva con el SI. Asimismo, a este elemento se le da una prioridad más alta que al elemento *Contemplar los potenciales errores* (*Alta* y *Media*, respectivamente), con el fin de resaltar su importancia por sobre la gestión y la manipulación de los errores en el SI. Como se mencionó anteriormente, los aspectos identificados por (Riegel y Doerr, 2015) se pueden refinar en subcategorías para apoyar la identificación y definición de estos elementos. En Scrum-UIA, estos elementos se obtuvieron y priorizaron en función de las exigencias y de las características de la tipología del proyecto, y se definieron y priorizaron en base a los resultados de las actividades 1.1, 1.2 y 1.3 de la Investigación Contextual.

Como se puede observar en la Tabla 21, los aspectos *Usabilidad*, *Contenido* y *Valor de Negocio* se componen de 6, 8 y 6 elementos, respectivamente. Asimismo, cada elemento tiene asignada una prioridad (*L*), utilizando la escala *Alta*, *Media* o *Baja*, computada con la fórmula (2).

Elementos de los Aspectos Relevantes del Proyecto			
Aspectos	Elementos	Prioridad	Prioridad (L)
Usabilidad	Proveer adaptabilidad al entorno de aprendizaje.	Alta	3
	Proveer facilidad de aprendizaje.	Alta	3
	Proveer accesibilidad a la información (opciones de formato).	Alta	3
	Reducir la carga mental de trabajo.	Media	2
	Contemplar los potenciales errores.	Baja	1
	Proveer facilidad del SI para que éste sea recordado.	Alta	3
Contenido	Datos personales de los estudiantes.	Media	2
	Contenido de las entidades de conocimiento.	Alta	3
	Datos personales de los instructores.	Baja	1
	Comunicaciones entre estudiantes e instructores.	Media	2
	Preferencias de los estudiantes.	Alta	3
	Descripción de las entidades de conocimiento.	Alta	3
	Retroalimentación de las actividades de evaluación.	Alta	3
	Registro de las actividades de los estudiantes.	Alta	3
Valor de Negocio	Gestionar el progreso de los estudiantes.	Medio	2
	Gestionar la matrícula.	Alta	3
	Monitorear el proceso de aprendizaje.	Alta	3
	Gestionar las entidades de conocimiento.	Alta	3
	Organizar el almacenamiento de la información.	Media	2
	Coordinar composición de las entidades de conocimiento.	Alta	3

**Tabla 21:** Elementos de los aspectos relevantes del SI de gestión de aprendizaje de la universidad.

Una vez identificados los aspectos y los elementos relevantes del SI de gestión de aprendizaje de la universidad, la siguiente tarea corresponde a la *priorización de los*

*requisitos en base a los elementos de los aspectos* (correspondiente a la Tarea 2.2.4 del Grupo de Actividades de la Gestión de Requisitos de Scrum-UIA). Esta tarea consiste en generar la priorización de los requisitos a través de la relación entre los elementos de los aspectos relevantes (ver Tabla 21) y los requisitos (ver Tabla 19). Para mayor detalle, revisar la sección 3.3.1.3. El resultado de esta tarea se presenta en la Tabla 22. Como se puede observar, mientras que en la primera y segunda columnas se listan los aspectos y los elementos del proyecto, en las otras columnas se identifican los diferentes requisitos del SI. De este modo, en la intersección de cada fila con cada columna se identifica si el elemento se encuentra relacionado con el requisito, identificándose con un 1, tal y como se define en la fórmula (3).

Priorización Basada en los Elementos de los Aspectos Relevantes del Proyecto															
Aspectos	Elementos	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14
Usabilidad	Proveer adaptabilidad al entorno de aprendizaje.	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	Proveer facilidad de aprendizaje.	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	Proveer accesibilidad a la información (opciones de formato).	0	0	0	1	0	0	1	1	0	0	0	0	1	1
	Reducir la carga mental de trabajo.	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	Contemplar los potenciales errores.	1	0	1	0	0	1	0	0	0	1	0	0	0	0
	Proveer facilidad del SI para que éste sea recordado.	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Contenido	Datos personales de los estudiantes.	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	Contenido de las entidades de conocimiento.	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	Datos personales de los instructores.	0	0	0	1	0	0	0	0	0	0	0	0	0	0

	Comunicaciones entre estudiantes e instructores.	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	Preferencias de los estudiantes.	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	Descripción de las entidades de conocimiento.	1	1	0	0	0	0	0	0	1	0	0	0	0	0
	Retroalimentación de las actividades de evaluación.	0	0	1	0	0	0	0	0	1	0	0	0	0	0
	Registro de las actividades de los estudiantes.	0	0	0	0	0	0	0	1	0	1	0	0	0	0
Valor de Negocio	Gestionar el progreso de los estudiantes.	0	1	0	0	0	0	0	0	0	0	0	0	1	0
	Gestionar la matrícula.	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	Monitorear el proceso de aprendizaje.	0	0	1	0	0	0	0	0	1	0	0	0	0	0
	Gestionar las entidades de conocimiento.	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	Organizar el almacenamiento de la información.	0	0	0	0	1	0	0	0	0	0	0	1	0	0
	Coordinar composición de las entidades de conocimiento.	1	0	0	0	1	1	0	0	1	0	0	1	0	0

**Tabla 22:** Priorización de los requisitos, basada en los elementos de los aspectos relevantes del SI.

Una vez generada la priorización de los requisitos de acuerdo a los elementos del proyecto, la última tarea consiste en la *generación del ranking final de los requisitos* (correspondiente a la Tarea 2.2.4 del Grupo de Actividades de la Gestión de Requisitos de Scrum-UIA). Esta tarea consiste en computar la clasificación final de los requisitos, de acuerdo a las relaciones establecidas en la tarea anterior (ver Tabla 22). Así pues, se calcula el nivel de relevancia de los requisitos por cada uno de los aspectos definidos, tal y como se define en la fórmula (8) y, a continuación, se genera la clasificación final de los requisitos, considerando el nivel de relevancia de cada requisito por aspecto, computada con la fórmula (12). Para mayor detalle, revisar la sección 3.3.1.4. En la Tabla 23, se presenta el ranking final de los requisitos del SI de gestión de aprendizaje de la universidad. Adicionalmente, se ha elaborado la Figura 19 para ilustrar, de manera gráfica, todo el proceso de priorización.



Como se puede observar, los requisitos se ordenan de acuerdo a su ranking final (*FR*), calculado con la fórmula (12). De esta manera, R2 y R12 son los requisitos con mayor y menor prioridad, respectivamente. Estos requisitos obtienen un ranking final de 12,17 y 2,67, respectivamente. Cabe destacar que el ranking final se encuentra sin colisiones de requisitos, es decir, cada requisito obtiene una clasificación final distinta.

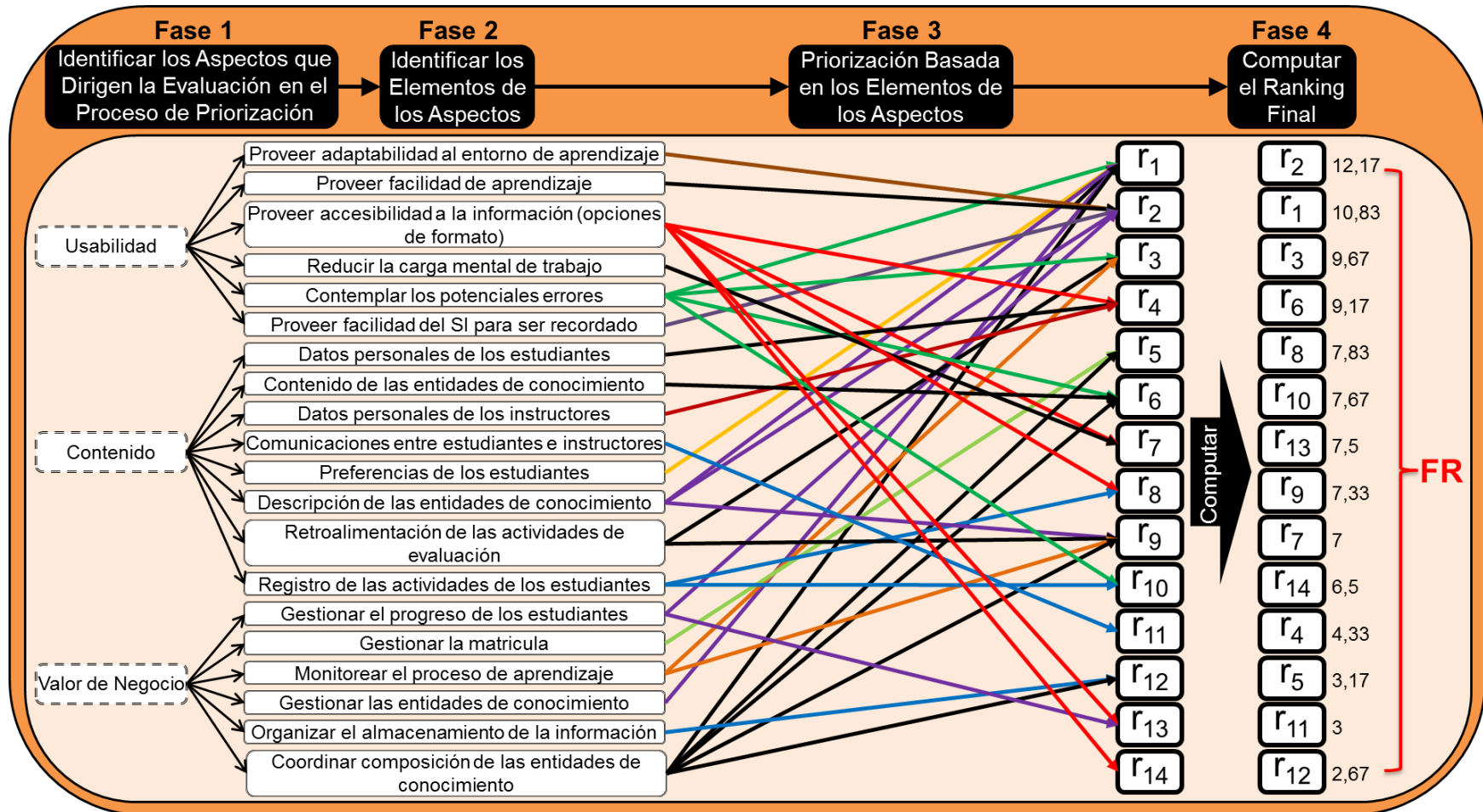
Requisitos del Proyecto	Ranking Final ( <i>FR</i> )
R2: Gestionar cursos en un entorno integrado	12,17
R1: Crear y editar cursos	10,83
R3: Crear y editar actividades de evaluación	9,67
R6: Incorporar material en los cursos	9,17
R8: Consultar información sobre el progreso de los estudiantes	7,83
R10: Enviar, de manera electrónica, las actividades de evaluación	7,67
R13: Crear aulas virtuales	7,5
R9: Consultar evaluaciones de las actividades	7,33
R7: Consultar los eventos y tareas de los cursos	7
R14: Revisar el SI como otro usuario	6,5
R4: Configurar perfiles de estudiantes y profesores	4,33
R5: Gestionar las inscripciones en los cursos	3,17
R11: Crear entornos de discusión	3
R12: Proveer un repositorio de archivos	2,67

**Tabla 23:** Ranking final de los requisitos del SI de gestión de aprendizaje de la universidad.

Asimismo, estos resultados también permiten observar la capacidad del método de priorización QMPSR para resolver problemas de colisiones de requisitos. Por ejemplo, los requisitos R3 y R6 están asociados con elementos que tienen la misma prioridad (ver Tabla 21 y Tabla 22). Es decir, cada uno de estos requisitos tiene asociado un elemento del aspecto *Usabilidad* con prioridad *Baja*, un elemento del aspecto *Contenido* con prioridad *Alta* y un elemento del aspecto *Valor del Negocio* con prioridad *Alta*. No obstante, los requisitos R3 y R6 obtienen una clasificación final (*FR*) diferente (9,67 y 9,17, respectivamente), debido a que el Factor de Asociación (*G*), computado con la fórmula (5), es distinto para cada elemento. A modo de ejemplo, se puede observar en la Tabla 22, que

los requisitos R3 y R6 tienen asociados los elementos “Retroalimentación de las actividades de evaluación” y “Contenido de las entidades de conocimiento”, respectivamente, ambos de prioridad *Alta* y pertenecientes al aspecto *Contenido*. Sin embargo, el elemento “Retroalimentación de las actividades de evaluación” tiene un mayor número de requisitos asociados (ver Tabla 22), generando, como resultado, que su prioridad sea más alta. Por lo tanto, como ya se mencionó en la sección 3.3.1.4, un elemento con un mayor número de requisitos asociados corresponde a un mayor *G* y, consecuentemente, su prioridad es más alta. Esto permite reducir los problemas de colisiones entre los requisitos asociados a elementos con la misma prioridad, generando resultados precisos en el ranking final de requisitos.

Por otro lado, una vez priorizados los requisitos (actividad 2.2), en la actividad 3.3 de Scrum-UIA, se seleccionan los requisitos que el Equipo de Desarrollo realizará en la Ejecución del Sprint, y que forman parte de la Lista de Pendientes del Sprint. El conjunto de requisitos seleccionados dependerá de la velocidad y la disponibilidad de recursos humanos que tendrá el Equipo de Desarrollo para el siguiente Sprint (definidas en la actividad 3.2). De este modo, por cada uno de los requisitos seleccionados, en la actividad 3.4, se identifican y definen las tareas específicas de desarrollo que se requieren para implementarlos. Posteriormente, en la actividad 3.5, se organiza la ejecución de estas tareas de desarrollo específicas, con el fin de fomentar su desarrollo incremental a través de la AI. Finalmente, durante la Ejecución del Sprint, se llevan a cabo las actividades del grupo de Inspección y Mejora Continua, donde se revisa el estado de las tareas de desarrollo (actividad 4.1), se inspecciona el incremento del producto generado por el Equipo de Desarrollo (actividad 4.2) y se reflexiona respecto al Sprint completado (actividad 4.3).



**Figura 19:** Priorización de requisitos mediante el método de priorización QMPSR.

### 3.3.3. Recapitulación

A lo largo de esta sección, se ha presentado una formulación completa del método de priorización propuesto para Scrum-UIA, incluyendo las definiciones y propiedades más importantes. Este método cualitativo para la priorización de los requisitos de software permite considerar, de manera formal, los aspectos y elementos relevantes, relacionados con la usabilidad, la AI y el valor de negocio. Estas prioridades se establecen en el panorama general de Scrum-UIA (detallados en la Tarea 1.4.1 del Grupo de Actividades de la Investigación Contextual), y se negocian dinámicamente entre el Dueño de Producto y el Arquitecto de la Información. Esto permite resolver las limitaciones de los métodos de priorización actuales, que se identifican en la sección 2.5, y corroborar la hipótesis de partida **H2.2**: Es posible idear un método de priorización que permita a la metodología anteriormente ideada (hipótesis de partida **H2.1**) conducir las prioridades de los requisitos a través de las preferencias de la usabilidad, de la Arquitectura de la Información y del valor de negocio, generando una clasificación final que permita reducir las colisiones de requisitos y presentar un buen comportamiento en términos de escalabilidad.

No obstante, para corroborar la hipótesis de partida **H2.2**, también se requiere evaluar el comportamiento de QMSPR en diferentes condiciones de rendimiento, con el fin de identificar su capacidad para reducir las colisiones de requisitos y presentar un buen comportamiento en términos de escalabilidad. Debido a lo anterior, en la sección 5.1, se detallarán los experimentos realizados para evaluar el método de priorización QMPSR frente a seis métodos de priorización existentes, con el objetivo de analizar en profundidad las colisiones de requisitos software generadas.

### 3.4. InterArch-T

Como se ha indicado en la descripción de Scrum-UIA (secciones 3.1 y 3.2), una de las características principales de esta metodología es que se centra en un modelo de desarrollo incremental e iterativo para entregar software usable rápidamente y tan a menudo como sea posible en un entorno cambiante. Asimismo, uno de los objetivos principales de Scrum-UIA es promover la implementación de los requisitos de forma incremental a través de la Arquitectura de la Información. Es por esto que, en esta sección se detalla la técnica ideada para promover el desarrollo incremental en Scrum-UIA a través de los entregables de la Arquitectura de la Información. Esta técnica se ha implementado mediante la construcción de una herramienta CASE, que permite al Arquitecto de la Información representar información conceptual de contenidos y generar diagramas que dan soporte al análisis y diseño de una aplicación interactiva. Esto permitirá corroborar la hipótesis de partida **H2.3**: Es posible promover el desarrollo incremental en la metodología anteriormente ideada (hipótesis de partida **H2.1**) a partir de las descripciones conceptuales que el Arquitecto de la Información crea, dentro del dominio del problema, y que evoluciona a fases más cercanas al dominio de la solución. Asimismo, es posible sistematizar esta técnica mediante la construcción de una herramienta CASE (Computer Aided Software Engineering), fácil de usar y de aprender por el Arquitecto de la Información.

En primer lugar, en la sección 3.4.1, se plantean los lineamientos generales de la técnica propuesta. En la sección 3.4.2, se describen los componentes arquitecturales principales de la técnica, mientras que en la sección 3.4.3, se describen las diferentes funcionalidades de la herramienta CASE InterArch. En la sección 3.4.4, se detalla el conjunto de reglas de transformación que dan lugar a la técnica para representar el conocimiento proveniente de las definiciones conceptuales del Arquitecto de la Información. Finalmente, en la sección 3.4.5, se describe un caso de uso, que permite profundizar el funcionamiento de InterArch.

#### 3.4.1. Lineamientos Generales

Como se indicó anteriormente, en Scrum-UIA se promueve el desarrollo incremental a través de los diferentes niveles de fidelidad de los entregables de la AI, con el fin de asegurar (durante la Reunión de Planificación del Sprint) que la implementación de los requisitos se dirige por las prioridades de la AI de una manera ágil y centrada en el usuario. No obstante, para llevar a cabo lo anterior, inicialmente se hace necesario identificar qué entregables de la AI son susceptibles de procesar automáticamente, para generar los diagramas de objetos que definen la aplicación en el dominio de la solución, y que facilitan el desarrollo incremental en Scrum-UIA.

Así pues, es posible resumir los entregables más comunes que el Arquitecto de la Información crea para llevar a cabo el análisis de la AI. Estos son los planos, las maquetas, los modelos de contenido y los vocabularios controlados (Erlin et al., 2008; Morville y

Rosenfeld, 2006), que se describen en la sección 2.2.3. Estos entregables proporcionan importantes conocimientos sobre el análisis, la organización, la gestión y la estructuración de la información, para los profesionales involucrados en el desarrollo de los proyectos de la Web. Sin embargo, entre todos estos entregables, los modelos de contenido son particularmente trascendentales para los miembros del Equipo de Desarrollo de Scrum-UIA (Analistas e Ingenieros del Software), ya que representan los aspectos no funcionales de las aplicaciones Web, y son susceptibles de procesarse automáticamente para generar los diagramas de clases y elementos de contenido, que definirán la aplicación en el dominio de la solución.

Por lo tanto, se ha diseñado una técnica, llamada InterArch-T (**Inter**operable Information **Arch**itecture **Technique**), que se ha implementado mediante la construcción de una herramienta CASE, llamada InterArch (**Inter**operable Information **Arch**itecture) (Rojas y Macías, 2013; Rojas y Macías, 2012; Rojas y Macías, 2011). Esta técnica tiene como objetivo general transformar la información de contenido no funcional, elaborada por el Arquitecto de la Información, en diagramas requeridos para el diseño de las aplicaciones Web, con el fin de proveer continuidad en la implementación de los requisitos en Scrum-UIA a través de la AI.

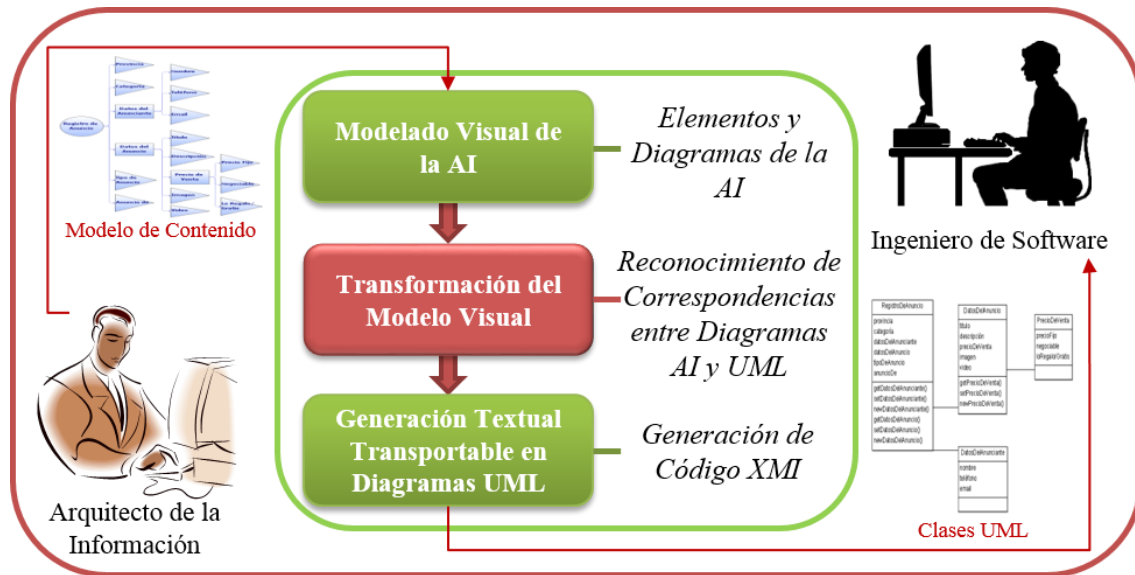
Más específicamente, la técnica InterArch-T se basa en dos principios esenciales. En primer lugar, dado que el Arquitecto de la Información, por lo general, tiene un perfil menos técnico, más orientado al diseño de información y organización, esta técnica permite que éste se concentre en tareas de análisis conceptuales en el dominio del problema. En segundo lugar, en base al análisis inicial realizado por el Arquitecto de la Información, esta técnica genera automáticamente diagramas UML para los miembros del Equipo de Desarrollo de Scrum-UIA, identificando los elementos que tienen su correspondencia con los diagramas de clases y objetos de contenido que utilizan los profesionales de software. Para ello, esta información se genera en un formato textual y transportable en XMI (XML, 1999), que es el estándar definido por la OMG para el intercambio de diagramas UML, de forma que sea procesable por cualquier herramienta CASE existente, con el objetivo de dar continuidad a los requisitos de manera incremental en el resto de fases y actividades de Scrum-UIA.

### 3.4.2. El Diseño Arquitectónico

Una vez descritos los lineamientos principales de la técnica InterArch-T, en esta sección se proveen los detalles arquitectónicos de esta técnica, así como de su herramienta InterArch, que permiten integrar la AI en entornos dinámicos y fomentar el desarrollo incremental en Scrum-UIA a través de la AI.

La técnica InterArch-T se compone de un conjunto de procesos que son responsables de la gestión y transformación de los modelos en un entorno visual, orientado al Arquitecto de la

Información. Como se muestra en la Figura 20, estos procesos incluyen: *el Modelado Visual de la AI*, *la Transformación del Modelo Visual*, y *la Generación Textual Transportable en Diagramas UML*. Estos procesos tienen como objetivo tomar como entrada el diseño visual de diagramas del Arquitecto de la Información, y generar como salida diagramas UML para los miembros del Equipo de Desarrollo de Scrum-UIA (Analistas e Ingenieros del Software).



**Figura 20:** Detalle arquitectónico de la técnica InterArch-T para promover el desarrollo incremental de Scrum-UIA y la integración de la AI en entornos ágiles.

La idea principal detrás de estos componentes de la arquitectura es permitir al Arquitecto de la Información trabajar en el modelado visual de una manera transparente, también incorporando una potente capa de interpretación que reconoce las diferentes correlaciones entre los diagramas de AI y las clases UML, requeridas por los Ingenieros del Software. La transformación del modelo visual se basa en un conjunto de relaciones y reglas de asociación que se aplican al modelo conceptual, producido por el Arquitecto de la Información, generando un conjunto de diagramas UML en un formato XML transportable, llamado XMI.

El Modelado Visual de la AI es el primer proceso que se muestra en la Figura 20 (de izquierda a derecha), que se lleva a cabo utilizando la interfaz de usuario principal de la herramienta InterArch. Tal interfaz es el entorno de trabajo principal del Arquitecto de la Información, y se compone de diferentes barras de herramientas que se utilizan para elaborar y gestionar los diagramas en un entorno funcional. En la Figura 21, se muestra esta interfaz de usuario, donde las partes principales están marcadas con letras mayúsculas (A, B y C). Este proceso permite al Arquitecto de la Información implementar los diferentes diagramas para la AI. El segundo proceso que se muestra en la Figura 20 es la

Transformación del Modelo Visual, que incluye la identificación de cada uno de los elementos visuales, producidos por el Arquitecto de la Información para componer, más adelante, los diagramas UML, utilizados por los miembros del Equipo de Desarrollo de Scrum-UIA (Analistas e Ingenieros del Software). Esto se logra a través de las reglas de asociación y relación, que se aplican a los elementos visuales individualmente o en grupos. Este proceso permite el reconocimiento de la relación entre los diagramas de la AI y los elementos de tipo UML (clases, operaciones, atributos, asociaciones, etc.).

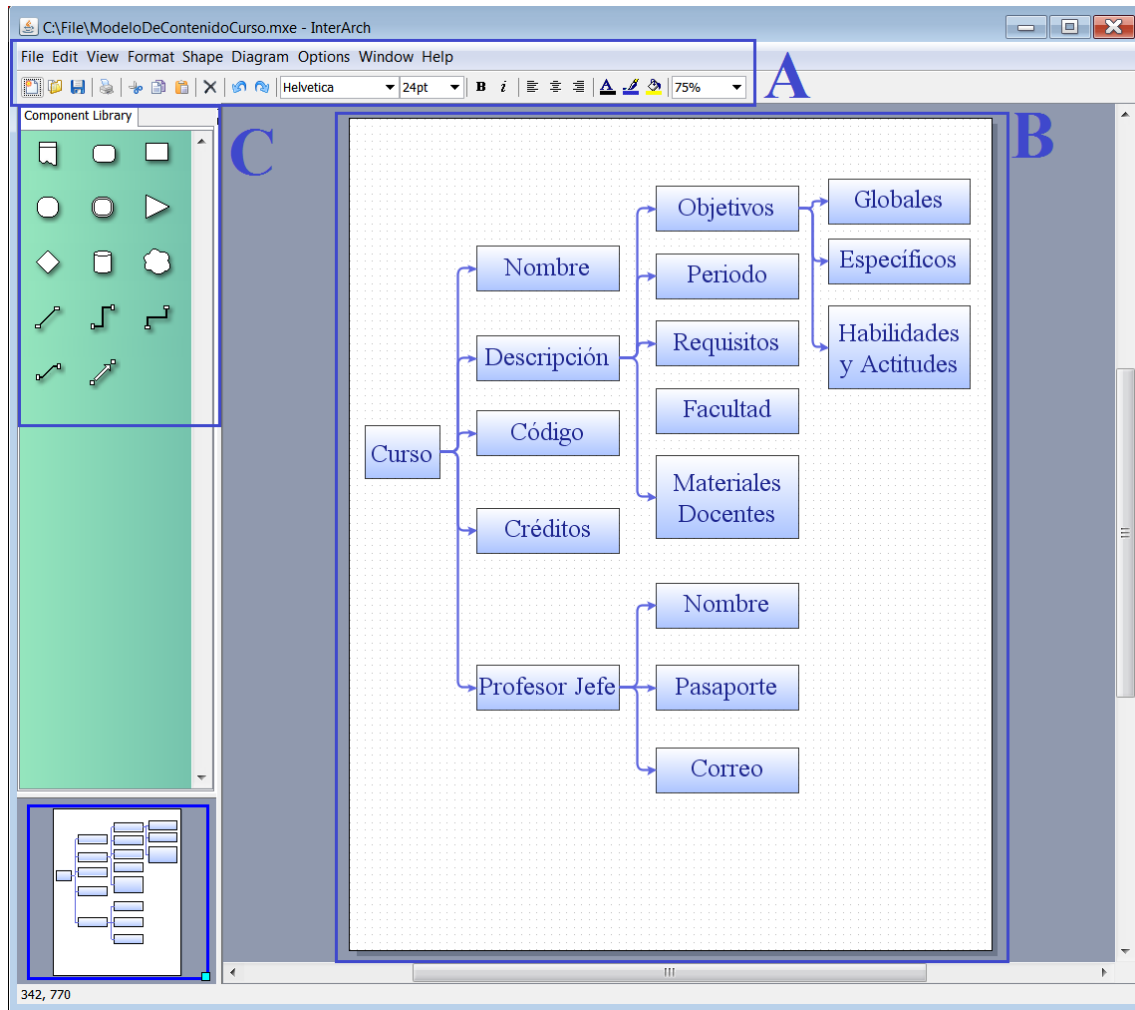
Finalmente, el último proceso que se muestra en la Figura 20 es la Generación Textual Transportable en Diagramas UML. Este proceso toma como entrada las relaciones creadas en la etapa de Transformación del Modelo Visual, y genera diagramas de clases UML en formato XMI para Ingenieros del Software. El objetivo es llevar a cabo el análisis y diseño de las fases en el proyecto y combinar estos diagramas con la parte funcional de la aplicación Web interactiva, a través de otras herramientas CASE, utilizadas durante el resto de las fases de desarrollo técnico del proyecto. De esta manera, los modelos de contenido, creados en el dominio del problema, evolucionan hacia modelos más cercanos al dominio de la solución (diagramas de clases UML en formato XMI), que se pueden utilizar por otros miembros del Equipo de Desarrollo de Scrum-UIA, para dar continuidad a la implementación de los requisitos de manera incremental.

### **3.4.3. Descripción de la Herramienta CASE InterArch**

Como ya se ha comentado, la técnica InterArch-T se ha implementado a través de la construcción de la herramienta CASE InterArch. Esta herramienta provee diferentes funcionalidades para elaborar las tareas de desarrollo (definidas por el Arquitecto de la Información y el Equipo de Desarrollo durante la Tarea 3.4.1 del Grupo de Actividades de la Planificación del Sprint), relacionadas con la AI durante la Ejecución del Sprint. Por lo tanto, en esta sección se detallan todas las prestaciones de InterArch, con el fin de identificar los elementos visuales que se proveen para elaborar diagramas de la AI, los estilos de manipulación disponibles y las opciones de formato y edición de los componentes. Asimismo, se indican las tecnologías que se utilizan en la programación de esta herramienta.

En general, la herramienta CASE InterArch (ver Figura 21) permite operar, formatear y vincular elementos visuales de contenido para producir diagramas de la AI. De esta manera, manipulando los elementos visuales, el profesional puede implementar los modelados visuales de la AI. Estas y otras facultades de la herramienta se explican en detalle en esta sección, algunas de las cuales corresponden al entorno de trabajo, a las opciones de formato y edición de los componentes, a los elementos visuales y a las diferentes formas para su manipulación:





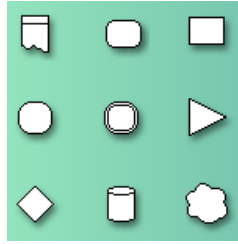
**Figura 21:** Interfaz de usuario de InterArch dividida en tres partes: A, B y C.

#### a) Elementos Visuales para Diagramas

La parte C de la Figura 21 expone los iconos de trabajo de los diferentes elementos visuales, enriqueciendo de esta manera la interpretación visual de los diagramas, realizados por el Arquitecto de la Información. Su tipología cuenta con dos tipos de elementos primordiales para el modelado visual:

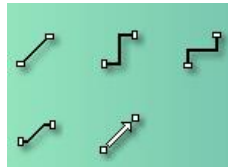
- *Los elementos para la interpretación de contenido del modelado visual-conceptual de la AI.* Estos son los elementos que permiten el manejo e interpretación de las entidades de contenido del modelado visual-conceptual de la AI. La Figura 22 presenta los elementos de contenido de la herramienta InterArch. A pesar de que los elementos de contenido tienen diferentes formas y estilos visuales, todos cuentan con la misma finalidad, es decir, permiten interpretar los objetos de contenido y datos en forma de diagrama. El Arquitecto de la Información ha de definir los objetos visuales que representarán a las entidades de contenido a su gusto, a través

de los elementos de contenido. Dichos elementos se deben trasladar al entorno de trabajo de la herramienta InterArch para su posterior ejecución.



**Figura 22:** Iconos de elementos de contenido.

- *Los elementos de enlace que permiten definir los tipos de asociaciones y relaciones entre los elementos de contenido.* La Figura 23 expone los elementos de enlace con los que cuenta la herramienta InterArch. Igual que en el caso de los iconos de la Figura 22, los elementos de enlace presentan diferentes formas y estilos visuales. No obstante, la finalidad de cada uno de ellos es la misma: permitir al Arquitecto de la Información establecer relaciones entre los diferentes elementos de contenido, permitiendo, a la vez, determinar el tipo de jerarquía de los diagramas. La manipulación de estos elementos de enlace requiere identificar el elemento de contenido tanto de origen como de destino.



**Figura 23:** Iconos de elementos de enlace.

Tal clasificación tiene como objetivo posibilitar el tratamiento apropiado en la aplicación de reglas de transformación, consiguiendo una generación adecuada de diagrama de clases UML.

#### **b) Entorno de Trabajo Visual**

Tal y como se ha indicado anteriormente, el entorno de trabajo visual corresponde esencialmente a la parte B de la Figura 21. Este entorno de trabajo permite manejar y vincular los distintos elementos visuales de la herramienta. El ejemplo en la Figura 21 ilustra las relaciones representadas entre los elementos de contenido, que describen la estructuración y la composición de cada curso de una universidad.

**Manipulación de elementos visuales:** El manejo de los elementos visuales dentro del entorno de trabajo de la herramienta InterArch requiere trasladarlos hacia el entorno de

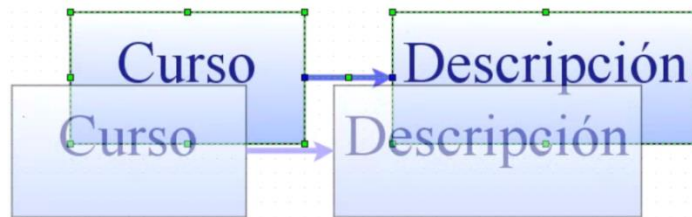
trabajo, con el fin de estructurar las distintas opciones con las que cuenta cada elemento visual. Algunas formas de manipulación con las que cuenta la herramienta InterArch son:

- Selección del elemento de contenido de inicio, arrastrándolo hacia el elemento de contenido de fin. De esta forma se genera un elemento de enlace entre los elementos de contenido. En la Figura 24, se pueden observar dos elementos de contenido (Curso y Descripción), conectados con un elemento de enlace.



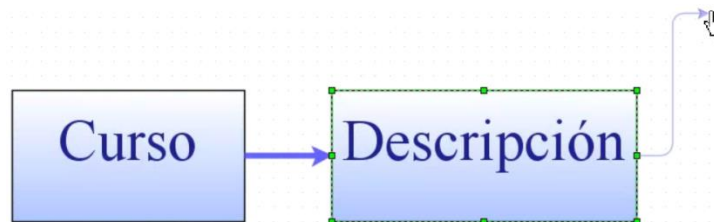
**Figura 24:** Manipulación para enlazar elementos de contenido en el entorno de trabajo.

- Los elementos del diagrama se pueden agrupar y manipular en bloque. Esto se logra a través de la selección de los elementos visuales, habilitando la agrupación o manipulación en bloque de todos los elementos visuales seleccionados. Esto se puede observar en la Figura 25, donde se manipulan en bloque los elementos de contenido Curso y Descripción, tratando de modificar la posición de estos elementos dentro del entorno de trabajo de la herramienta InterArch.



**Figura 25:** Manipulación en bloque de los elementos en el diagrama.

- Los elementos tienen propiedades de herencia dentro del entorno de trabajo. Es decir, es posible generar nuevos elementos de contenido para cualquier elemento. Por lo tanto, se crean nuevos elementos de contenido con las propiedades del elemento de contenido de origen. Para llevarlo a cabo es necesario posicionarse sobre el elemento de contenido, seleccionar el elemento y arrastrarlo hacia la posición deseada (cambiando el cursor a un formato de *mano*).



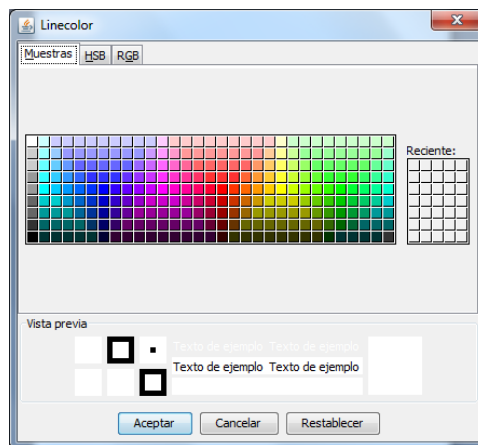
**Figura 26:** Creación de elementos visuales heredados.

En la Figura 26, se puede observar la creación de elementos visuales heredados: los elementos de contenido *Curso* y *Descripción* están asociados a través de un elemento de enlace, tratando de crear elementos visuales heredados desde el elemento de contenido *Descripción*.

### c) Formato y Edición de Componentes

En la parte A de la Figura 21, se muestran las opciones de formato y edición frecuentes de una aplicación de estas facultades para manipular elementos dentro del entorno de trabajo: vistas del entorno de trabajo, edición, archivos, formato y estilos, entre otros. Las opciones de archivo permiten controlar el formato, almacenamiento, así como la impresión de la página del diagrama. Las opciones de edición posibilitan la manipulación de edición de textos y recuperación de acciones en el entorno de trabajo visual de InterArch. En cuanto a las opciones de vistas, éstas generan diferentes opciones visuales del entorno de trabajo, mientras que las opciones de formato posibilitan configurar la forma, tamaño, estilo y comportamiento de los elementos visuales. Así pues, todas las opciones de formato y edición de los elementos visuales permiten configurar cada elemento visual tanto de forma individual como grupal, definiendo y manipulando los diagramas. Algunos ejemplos de este tipo de facultades de la herramienta InterArch se detallan a continuación:

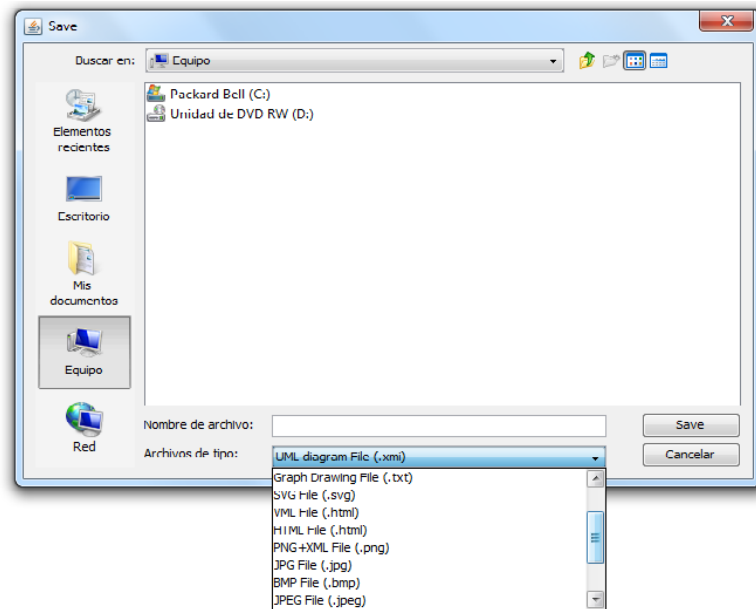
- La configuración del color se puede establecer para cualquier elemento visual de la herramienta InterArch. Esto es ilustrado en la Figura 27, que brinda un ejemplo de las opciones de color del formato de línea de los elementos visuales de la herramienta InterArch.



**Figura 27:** Paleta de colores de formato de línea.

- También existen diferentes formatos para almacenar los diagramas realizados, que se encuentran dentro de las opciones para almacenar el archivo. El cuadro de diálogo para almacenar los diagramas aparece en la Figura 28, donde se pueden apreciar los distintos formatos disponibles (.xmi, .txt, .svg, .html, .png, .jpg, .bmp,

.jpeg, .gif). Asimismo, se encuentra la opción de *UML diagram File (.xmi)*, que permite transformar el modelo de contenido, producido por el Arquitecto de la Información, en diagramas de clases UML en formato XMI. Esto implica procesar el modelo de contenido mediante la aplicación del conjunto de reglas de transformación proyectado (para mayor información, ver la sección 3.4.4), generando como consecuencia un archivo con extensión .xmi. Los otros formatos incluyen los de imagen (.png, .jpg, .bmp, .svg, .jpeg, .gif), páginas HTML (.html) y archivo de texto (.txt).

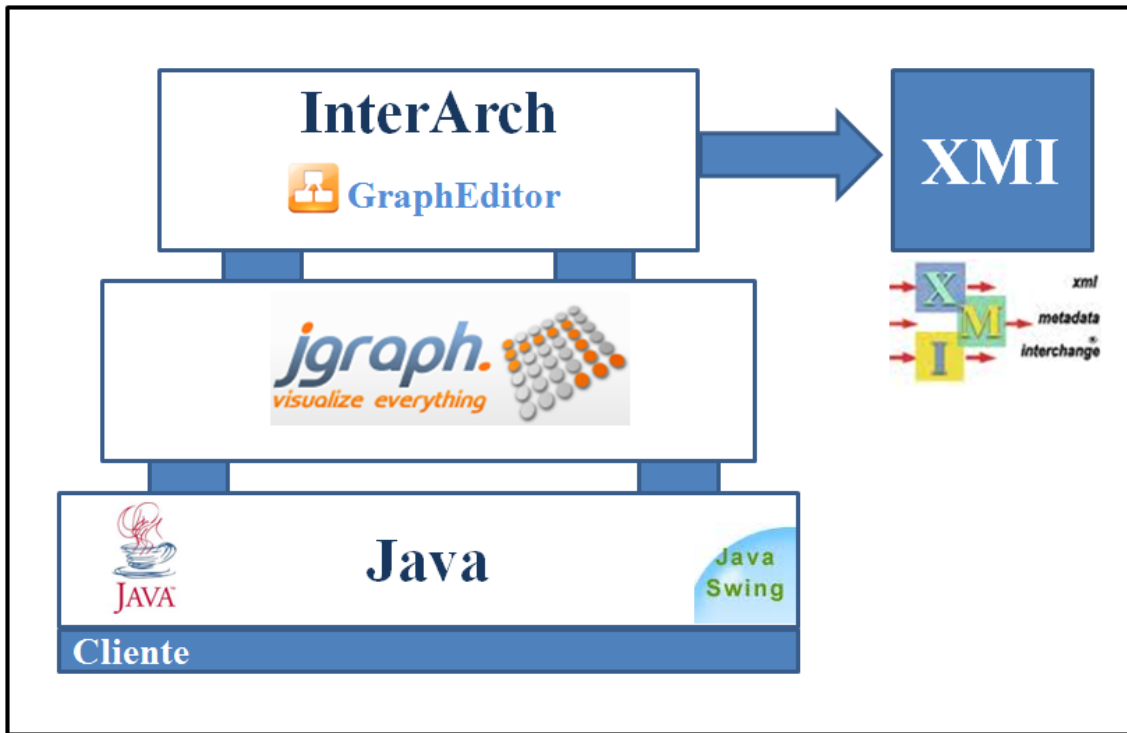


**Figura 28:** Formatos para almacenar el diagrama.

#### 3.4.3.1. Descripción de las Tecnologías Utilizadas

Una vez descritas las funcionalidades de InterArch para integrar la AI, en esta sección se describen las tecnologías aplicadas en la programación de esta herramienta.

En la Figura 29, se presenta el conjunto de tecnologías utilizadas. Como se puede observar, el *cliente* de Java es la base de todas las tecnologías empleadas, que se puede ejecutar en cualquier sistema operativo que soporte el uso de una Máquina Virtual de Java (JVM) (Lindholm y Yellin, 1999). Para diseñar la parte gráfica, se ha utilizado la librería JGraph, que se apoya en la tecnología Java. La herramienta InterArch fue desarrollada con la librería JGraph que se sitúa en la parte superior de la pirámide tecnológica. Finalmente, para la generación de diagramas de clases UML, se ha utilizado el formato textual y transportable XMI. Cada una de estas tecnologías se detallará a continuación.



**Figura 29:** Tecnologías utilizadas en la implementación de InterArch.

#### a) JGraph

Es un componente software de código abierto en el lenguaje de programación Java para diseñar gráficos. JGraph (David y Gaudenz, 2010) está basado en la teoría matemática de grafos, permitiendo fundar aplicaciones Java Swing con funcionalidad interactiva de diagramas. InterArch utiliza la versión 1.5.1.7 de esta librería, con el fin de desarrollar diagramas interactivos, basados principalmente en la manipulación de Vertex (unidad fundamental de la que están formados los grafos) y Edge (una relación entre dos vértices de un grafo). En concreto, la construcción de la InterArch se basa en la versión 1.8 de GraphEditor, permitiendo la edición de gráficos, siendo una implementación de la librería JGraph. InterArch ha incorporado funcionalidades en GraphEditor para las reglas de transformación (Jerarquía y Configuración) y ha modificado aspectos relacionados con el formato de la herramienta.

#### b) Java

Java fue desarrollado por Sun Microsystems a principios de los años 90, y tiene un lenguaje de programación orientado a objetos. Java fue lanzado con el fin de desarrollar un lenguaje específico de redes computacionales heterogéneas, independiente de la plataforma de ejecución. Además, un programa de Java puede ejecutarse en cualquier máquina o

plataforma (Gosling et al., 2005). Es precisamente el lenguaje de programación que se utiliza como soporte para desarrollar la herramienta InterArch.

#### c) XMI

XMI (XML de Intercambio de Metadatos) es un formato para el intercambio de diagramas, permitiendo expresar objetos con XML (lenguaje de marcas extensible). XMI está estrechamente ligado a los estándares de modelado y, por lo tanto, posibilita la eficacia de los modelos. Grose et al. (2002) identificaron los siguientes factores como los más valiosos de XMI:

- XMI proporciona una representación estándar de objetos en XML, permitiendo el intercambio efectivo de los objetos que utilizan XML.
- XMI especifica cómo crear esquemas XML a partir de modelos.

InterArch utiliza XMI como formato textual transportable de los diagramas de clases UML generados. La aplicación de este código no fue mediante el uso de librerías ad-hoc de procesamiento XMI (DOM, SAX, JOB, XMI Framework, entre otras), sino en base a un desarrollo propio. Cuando InterArch genera el archivo XMI, éste se puede procesar por herramientas CASE de modelado que soporten la importación de este formato.

#### d) Swing GUI Components

Java suministra la librería de componentes visuales Java Swing (Walrath y Campione, 1999), que incluye todo tipo de componentes, desde botones hasta paneles para dividir las tablas. Muchos de ellos son capaces de arrastrar, imprimir, clasificar y solar, entre otras cosas. Esta librería permite adaptar las interfaces de usuario de InterArch a cada sistema operativo, sin necesidad de cambiar el código fuente.

### 3.4.4. Reglas de Transformación

Como ya se comentó anteriormente, el objetivo general de la técnica InterArch-T es proporcionar información de análisis y diseño procesable por los miembros del Equipo de Desarrollo de Scrum-UIA, a partir de las descripciones conceptuales que elabora el Arquitecto de la Información. Para la generación de esta información de análisis y diseño, en la técnica InterArch-T, se incluye una capa de interpretación que comprende un conjunto de reglas de transformación. Estas reglas analizan las relaciones de asociación y jerarquía en los modelos de contenido, desarrolladas por el Arquitecto de la Información, con el fin de transformarlas en código UML. Las reglas de transformación se dividen en reglas de *Jerarquía* y de *Configuración*:

#### 3.4.4.1. Reglas de Jerarquía

Con el fin de contar con criterios formales para validar el procesamiento automático de los modelos de contenido, y la subsiguiente transformación en diagramas de clases UML, se ha definido un conjunto de reglas. Las Reglas de Jerarquía se ocupan de la estructura y de la jerarquía de los diagramas de modelo de contenido, producidas por el Arquitecto de la Información. Tales reglas se han inspirado en el modelo de procesos que aparece en (Pressman, 2005), proponiendo la transformación de la información de contenido no funcional, desarrollada inicialmente por los analistas, que se incluirá más adelante en las clases de aplicación funcional para el diseño de aplicaciones Web, con el fin de asegurar su continuidad en el diseño de los elementos no funcionales, que se lleva a cabo en las primeras fases del proceso de desarrollo Web. Más específicamente, en esta tesis doctoral se ha formalizado, mejorado, implementado y evaluado este enfoque anterior por medio de una herramienta CASE (InterArch), en base a un conjunto de reglas que permiten establecer correspondencias entre modelos de contenido, elaborados por Arquitectos de la Información y diagrama de clases UML, requeridos por Analistas e Ingenieros del Software. Además, se han incorporado nuevas características respecto a la configuración de los métodos, relaciones y reglas en la generación de los diagramas de clases UML.

Estas reglas se aplican a todos los elementos en el modelo de contenido para realizar las transformaciones de diagramas de clases UML correspondientes. Es decir, las reglas consideran la estructura y la jerarquía del contenido para crear clases, atributos, operaciones y asociaciones en los diagramas de clases UML resultantes.

En concreto, se incluyen cinco reglas de jerarquía principales que se pueden definir de la siguiente manera:

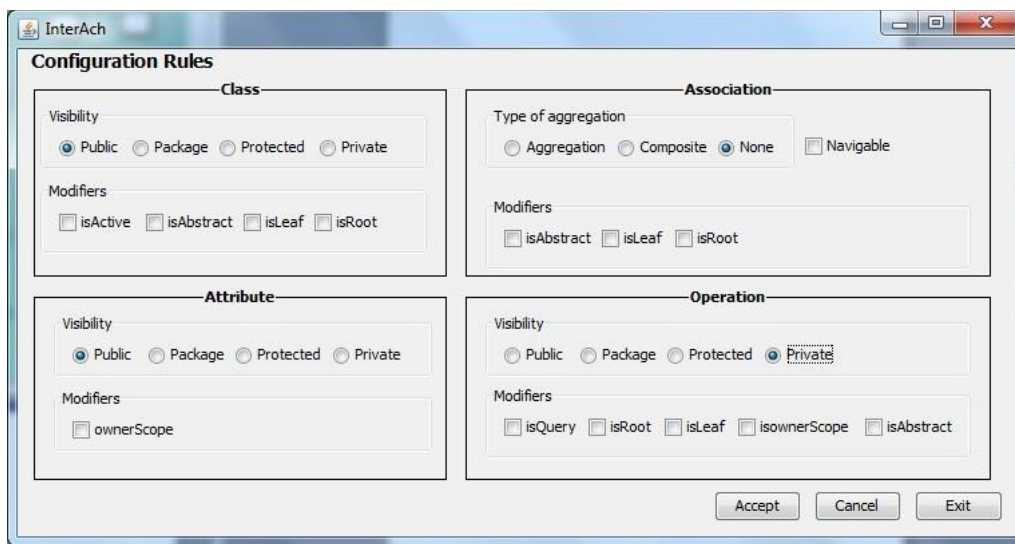
- **R1:** Un elemento de contenido, que contiene a otros elementos descendientes, se considera de una clase UML.
- **R2:** Un elemento de contenido descendiente se considera como un atributo, que se incluye en los elementos de la clase de la que desciende.
- **R3:** El elemento principal del diagrama de modelo de contenido será la clase principal en el diagrama de clases UML.
- **R4:** Un elemento de contenido descendiente, correspondiente a una nueva clase, genera una asociación directa con el elemento del que desciende.
- **R5:** Para cada una de las asociaciones generadas en el diagrama de clases UML, se crean tres métodos (get, set y new) y se incluyen en la clase de origen.

#### 3.4.4.2. Las Reglas de Configuración

Las reglas de configuración son un conjunto de propiedades específicas en relación con el nivel de visibilidad, accesibilidad y navegabilidad en las clases, atributos, métodos y asociaciones de los diagramas de clases UML. A diferencia de las reglas de jerarquía, las



reglas de configuración no consideran la estructura de los diagramas de contenido, producida por el Arquitecto de la Información. Por el contrario, las reglas de configuración se ocupan de propiedades que afectan a la generación de diagramas de clases UML. Otra diferencia con respecto a las reglas de jerarquía es que las reglas de configuración se centran principalmente en los Ingenieros del Software, debido a los conocimientos técnicos necesarios para la manipulación de propiedades UML. Esto significa que los Arquitectos de Información pueden utilizar InterArch para llevar a cabo el modelado del contenido en el espacio del problema, mientras que los Ingenieros del Software pueden utilizar la facilidad de la regla de configuración para especificar el UML, con el fin de adaptar mejor los requisitos de diseño en el espacio de la solución.



**Figura 30:** Módulo de InterArch para las reglas de configuración.

La Figura 30 muestra el módulo InterArch para la gestión de reglas de configuración. Además, la figura muestra las reglas seleccionadas de forma predeterminada, que se agrupan en las reglas de clase, de atributo, de método y de asociación. Estas reglas de configuración corresponden a los tipos de etiquetas, propuestos por la OMG (OMG, 2005), para adaptar los esquemas y documentos que se producen utilizando XMI. Las categorías de las reglas de configuración son las siguientes:

- **Clase:** Este grupo proporciona opciones para configurar la visibilidad (*public*, *package*, *protected* y *private*) y el tipo de acceso (*active*, *abstract*, *leaf* y *root*) para las clases. Está configurado por defecto que los atributos de clase tengan una visibilidad pública y sin ningún tipo de acceso definido.
- **Atributo:** Este grupo proporciona opciones para configurar la visibilidad (*public*, *package*, *protected* y *private*) y el tipo de acceso (*owner*

scope) para los atributos de clase. De forma predeterminada, los atributos tienen una visibilidad pública y no tienen acceso definido.

- **Asociación:** Este grupo comprende la configuración de las diferentes características, como la navegabilidad entre las clases, la agregación (aggregation, composite y none) y el acceso (root, leaf y abstract) para todas las relaciones de UML en el diagrama de clase. Por defecto, las asociaciones son estándar y no tienen un acceso definido o una navegación específica entre las clases.
- **Operación:** Este grupo proporciona opciones para configurar la visibilidad (public, package, protected y private) y el acceso (query, root, leaf, owner scope y abstract) para las operaciones de clase. De forma predeterminada, las operaciones tienen visibilidad pública y sin acceso definido.

### 3.4.5. Caso de Uso

Una vez descritas la técnica InterArch-T y su herramienta InterArch, en esta sección se provee un caso de uso específico, con el fin de mostrar en detalle la idoneidad de la InterArch-T para integrar la AI y dar continuidad al resto de actividades durante la Ejecución del Sprint en Scrum-UIA.

#### 3.4.5.1. Descripción de la Tarea para el Caso de Uso

Se trata de un caso de uso, en el que el Arquitecto de la Información desea trabajar en un modelo de contenido, durante la Ejecución del Sprint de Scrum-UIA, para representar la información de la ficha de un curso de una universidad. Esta ficha se muestra en la Figura 31, que representa una “maqueta” de fidelidad media, adquirida en la fase de obtención de requisitos de Scrum-UIA con el personal administrativo de la universidad. Como se puede observar, esta maqueta contiene información referente al nombre, código y créditos del curso. También incluye información sobre los requisitos y la facultad, e información respecto a los objetivos y materiales docentes del curso. Por lo tanto, el Arquitecto de la Información podría llevar a cabo el modelado conceptual, logrando las siguientes tareas:

- a) **Segmentación:** La primera tarea es la segmentación de contenido, utilizando la maqueta proporcionada en la Figura 31. Esta tarea identifica diferentes componentes y estructuras de información, obteniendo la composición y la jerarquía de los diversos elementos de información, contenida en la maqueta. Los resultados de la segmentación del contenido permiten que el Arquitecto de la Información tenga la información necesaria para preparar el modelo de contenido mediante el uso de InterArch.
- b) **Modelado de contenido:** Éste se lleva a cabo mediante el uso de InterArch, teniendo en cuenta la información previamente obtenida durante la segmentación, que genera un diagrama de información jerárquica, como se muestra en la parte B

de la Figura 21, donde el objeto de contenido *Curso* es descrito por otros cinco diferentes objetos de contenido (*Nombre*, *Código*, *Créditos*, *Descripción* y *Profesor Jefe*). *Descripción* y *Profesor Jefe* son datos compuestos, que son definidos según los objetos de contenido que jerárquicamente descienden de ellos. Una vez creado, el modelo de contenido se puede transformar automáticamente en diagramas UML, mediante la activación de las correspondientes reglas de transformación. Este paso es transparente para el Arquitecto de la Información y está dirigido con sólo haciendo clic en la opción de menú correspondiente.

<b>Curso</b>	
Nombre	Psiquiatría y Psicología Médica
Código	89810205
Créditos	21
<b>Descripción</b>	
Facultad	Medicina
Periodo	Primer trimestre
Requisitos	Historia de la Medicina y Documentación Médica
<b>Objetivos</b>	
Globales	Aplicar a la labor clínica los conocimientos psiquiátricos para un mejor ejercicio de la actividad profesional.
Específicos	Desarrollar habilidades y actitudes que precisa un licenciado en Medicina.
Habilidades y Actitudes	Ser capaz de dirigir la entrevista y exploración psicopatológica de un paciente con un trastorno mental.
<b>Materiales Docentes</b>	
Recursos	La enfermedad psiquiátrica, bulimia nerviosa.
Libros	Lecciones de psiquiatría.
<b>Profesor Jefe</b>	
Nombre	Juan Perez
Pasaporte	P050258740
Correo	juan.perez@correo.com

**Figura 31:** Maqueta de registro de cursos para una universidad.

### 3.4.5.2. Las Reglas de Asociación y de Relación para Procesar los Elementos Visuales

Para generar los diagramas UML desde el modelo de contenido de la AI, se aplican las reglas explicadas anteriormente. Esto se consigue al guardar el modelo de contenido en el disco, usando la opción “UML diagram File (.xmi)”. En este caso, la clase principal se genera a partir del elemento principal (*Curso*). Además, los elementos descendientes

generan las siguientes clases: los elementos finales y directos, descendientes del elemento principal, se transforman en atributos en la clase `Curso`. Si estos elementos son ambos descendientes, entonces se transforman en nuevas clases, directamente relacionadas con el elemento `Curso`. Si los elementos descendientes generan atributos y son elementos compuestos, entonces nuevas clases se generan de forma recursiva y en relación al elemento del que descienden. La aplicación de esta regla generaría un componente de clase que contiene cinco atributos: `nombre`, `codigo`, `creditos`, `descripcion` y `profesorJefe`. Además, aplicando las reglas, el elemento `profesorJefe` desciende directamente desde el elemento principal, y al mismo tiempo contiene los elementos descendientes `Nombre`, `Pasaporte` y `Correo`, por lo que se transforma en una nueva clase, mientras que sus elementos descendientes `Nombre`, `Pasaporte` y `Correo` en atributos de la clase `profesorJefe`. En cuanto a los métodos de las clases, tres métodos se generan por defecto para cada atributo que representa una clase agregada (`get`, `set` y `new`).

### 3.4.5.3. Generación de Código XML

La ejecución de las reglas genera automáticamente un diagrama de clases UML en un formato textual XMI. El siguiente fragmento de código representa el código XMI, generado para la clase `Curso`, que contiene algunos atributos, métodos y relaciones, relacionados con la clase `profesorJefe`, de acuerdo con el ejemplo anterior y el diagrama de salida, representado en la Figura 32.

```
<UML:Class xmi.id = 'x232' name = 'Curso'>
<UML:Classifier.feature>

<UML:Attribute xmi.id = 'x232:87B' name = 'nombre' visibility =
'public'></UML:Attribute>...

<UML:Operation xmi.id = 'x232:02C' name = 'getDescripcion'
visibility = 'public'></UML:Operation>...

<UML:Class xmi.id = 'x235' name = 'profesorJefe'>...

<UML:Association xmi.id='868'> <UML:Association.connection>
<UML:AssociationEnd xmi.id='889' aggregation='aggregate'>...

<UML:AssociationEnd.participant><UML:Class
xmi.idref='x232' /></UML:AssociationEnd.participant></UML:Associati
onEnd><UML:AssociationEnd xmi.id = '874'
aggregation='none'><UML:AssociationEnd.participant><UML:Class
xmi.idref='x235'>...
```

De esta manera, la clase `Curso` está representada por la etiqueta `<UML:Class>`. Además, el atributo `nombre` en la clase `Curso` está representado por la etiqueta `<UML:Attribute>`. En cuanto a los métodos de la clase, el método `getDescripcion`

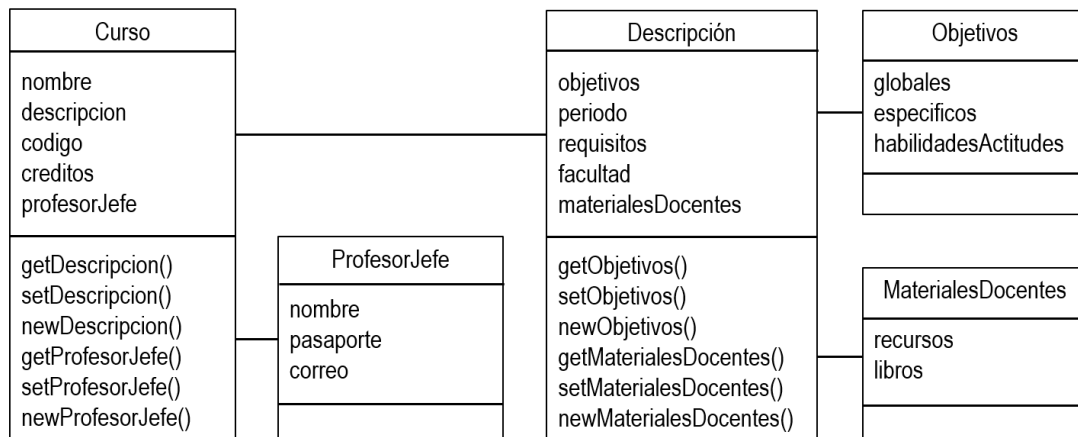
en la clase `Curso` está representado por la etiqueta `<UML:Operation>`. Todas estas etiquetas contienen un identificador único y el nombre de la etiqueta.

En cuanto a las asociaciones entre las clases en el diagrama de clases XML, éstas están representadas por `<UML:Association>`, junto con las etiquetas `<UML:AssociationEnd.participant>` y `<UML:AssociationEnd>`, que permiten especificar el tipo de asociación (*aggregation*, *composite* y *none*) y las clases que participan en la asociación. Por último, todas las etiquetas que definen las clases, los atributos, los métodos y las asociaciones tienen propiedades específicas que se pueden personalizar, como se ha descrito anteriormente, por el módulo de reglas de configuración de InterArch.

El archivo generado en formato XML es portátil y puede utilizarse en cualquier herramienta UML que soporte XML, para citar algunas de ellas: ArgoUML, StarUML, BOUML, VisualParadigm, Circa y Mia-generacional, entre otras.

#### 3.4.5.4. Diagrama de Clases UML

La Figura 32 representa el diagrama final de clases UML, generado automáticamente por InterArch. El diagrama de clases consiste en una clase raíz `Curso`, que contiene cinco atributos: *nombre*, *descripcion*, *codigo*, *creditos* y *profesorJefe*. Los métodos creados para esta clase, que denota las relaciones con clases *descripcion* y *profesorJefe*, son *getDescripcion*, *setDescripcion*, *newDescripcion*, *getProfesorJefe*, *setProfesorJefe* y *newProfesorJefe*. A su vez, la clase `Descripcion` está directamente relacionada con las clases de `Objetivos` y `MaterialesDocentes`.



**Figura 32:** Diagrama de clases UML obtenido desde la transformación del modelo de contenido de la Arquitectura de la Información.

Además, para una solución de diseño específica, podría ser necesario concretar la cardinalidad o composición de las relaciones (*strong aggregation*). Este es el caso

de las clases de `ProfesorJefe` y `Descripcion`, ambas relacionadas con la clase `Curso`, ya que no tiene sentido que `Descripcion` exista sin `Curso` (es decir, no tiene ningún sentido que la parte exista sin el conjunto). El tipo y nivel de dependencia entre las relaciones, así como la cardinalidad y navegabilidad, se pueden modificar por los Analistas o Ingenieros del Software, por medio de las reglas de configuración explicadas anteriormente.

En general, la información creada por InterArch se puede adaptar a las necesidades más específicas, importando los archivos XMI, generados en otras herramientas CASE. Esto ayuda a tomar ventaja de las características proporcionadas por otras herramientas para tratar con código de UML, como la ingeniería inversa, la integración de bases de datos y restricciones OCL, entre otros.

Finalmente, la aplicación del caso de uso anterior se centró en el modelado conceptual en el dominio del problema, y su posterior transformación en un modelo del dominio cercano a la solución permite corroborar la hipótesis de partida **H2.3**, es decir, la técnica InterArch-T (así como su implementación mediante la construcción de la herramienta CASE InterArch) permite promover el desarrollo incremental en Scrum-UIA, mediante la representación inicial de los modelos de contenido (elaborados por el Arquitecto de la Información), que evolucionan a diagramas de clases UML, requeridas por Analistas e Ingenieros del Software.

Sin embargo, se hace también necesario proporcionar características de facilidad de uso y aprendizaje en la manipulación de la herramienta InterArch. Por lo tanto, en la sección 5.2, se llevará a cabo un experimento con usuarios reales, para determinar el grado de usabilidad de la herramienta CASE InterArch, lo que permitirá finalmente corroborar la hipótesis de partida **H2.3**.

### 3.5. Scrum-UIA-MAT

En la metodología Scrum-UIA, se introducen diferentes prácticas y recomendaciones, con el fin de incluir la perspectiva de los usuarios finales durante todo el ciclo de vida del desarrollo de software. Asimismo, se proponen tareas y actividades para capturar las prioridades del proyecto desde las fases iniciales, que permiten guiar el proceso de priorización de los requisitos. No obstante, la mayoría de estas características son específicas de Scrum-UIA, por lo que no cuentan con un soporte tecnológico directo de las herramientas comerciales que comúnmente se utilizan para la gestión de proyectos software. Es por esto que, se ha implementado la metodología Scrum-UIA, así como su método de priorización (QMPSR), mediante la construcción de una herramienta Web, llamada Scrum-UIA-MAT (**Scrum-UIA MAnagement Tool**). Esta herramienta provee un sistema de gestión de proyectos, basado en Scrum-UIA, que considera diferentes funcionalidades para apoyar el Diseño Centrado en el Usuario y el desarrollo ágil de la AI.

Asimismo, la construcción de esta herramienta permite corroborar la última hipótesis de partida **H2.4**: Es posible crear una herramienta Web, que proporcione un sistema de gestión de proyectos basado en la metodología anteriormente ideada (hipótesis de partida **H2.1**), mediante un proceso ingenieril centrado en el usuario. De este modo, la herramienta Web desarrollada será fácil de usar y de aprender por parte de sus usuarios potenciales.

En primer lugar, en la sección 3.5.1, se presentan los requisitos funcionales y no funcionales de la herramienta Scrum-UIA-MAT. En la sección 3.5.2, se describen los componentes tecnológicos utilizados en su implementación. Finalmente, en la sección 3.5.3, se describe en detalle cada uno de los módulos de Scrum-UIA-MAT.

#### 3.5.1. Análisis

La herramienta Scrum-UIA-MAT tiene como objetivo configurar y gestionar los proyectos software que son implementados con Scrum-UIA. Por un lado, la arquitectura general (diseño estructural y disposición de los elementos) de Scrum-UIA-MAT se ha inspirado en herramientas similares que permiten organizar y gestionar los diferentes aspectos de un proyecto software. Por otro lado, los requisitos funcionales de esta herramienta se han obtenido considerando el modelo de proceso de Scrum-UIA, así como sus características y prácticas específicas. Estos requisitos funcionales corresponden a:

- **Gestión de los elementos y aspectos relevantes del proyecto:** El Dueño del Producto y el Arquitecto de la Información deben ser capaces de manipular (crear, consultar, modificar y eliminar) los elementos y los aspectos relevantes que permiten identificar las prioridades del proyecto. Asimismo, el Dueño del Producto y el Arquitecto de la Información han de diferenciar el nivel de importancia, tanto de los elementos como de los aspectos del proyecto.

- **Gestión de los usuarios finales del proyecto:** El Dueño del Producto y el Arquitecto de la Información deberían ser capaces de identificar y gestionar los usuarios finales del Sistema de Información, y relacionarlos con los requisitos del proyecto.
- **Gestión de los requisitos del proyecto:** El Dueño del Producto y el Arquitecto de la Información podrán registrar y gestionar los requisitos de cada proyecto. Cada requisito podrá contener información relacionada con el esfuerzo estimado, la descripción, los criterios de aceptación de los usuarios finales y los elementos relevantes relacionados, entre otros.
- **Visualización de la Lista de producto:** Todos los miembros del equipo Scrum-UIA tendrán disponibles diversas alternativas para consultar y recuperar los datos relacionados con la Lista del Producto.
- **Gestión de los Sprints del proyecto:** El Equipo de Desarrollo podrá configurar las características generales (objetivo, duración, descripción, entre otros) de cada Sprint del proyecto. Asimismo, los otros integrantes del equipo Scrum-UIA tendrán acceso para consultar los diferentes datos de configuración de los Sprints.
- **Gestión de las tareas de desarrollo:** El Equipo de Desarrollo tendrá opciones para gestionar las tareas de desarrollo, asociadas a cada requisito del Sprint.
- **Consultar las tareas de desarrollo de los Sprints:** Todos los miembros del equipo Scrum-UIA podrán consultar las tareas de desarrollo que son asignadas en cada Sprint del proyecto.
- **Planificación de las tareas de desarrollo:** El Equipo de Desarrollo podrá detallar el esfuerzo estimado para cada día de las tareas de desarrollo.

Asimismo, en la herramienta Scrum-UIA-MAT, también se deben proveer cualidades de facilidad de uso y de aprendizaje por parte de sus usuarios potenciales. Por lo tanto, se ha definido un conjunto de requisitos no funcionales, con el objetivo de incorporar aspectos de usabilidad. A continuación, se describen los requisitos no funcionales de la herramienta Scrum-UIA-MAT:

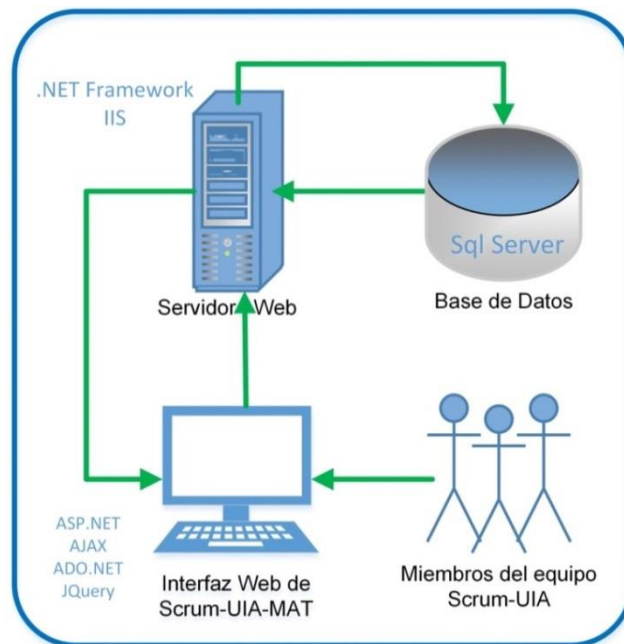
- **Facilidad de uso:** Las opciones de la herramienta deberán diseñarse para reducir la carga cognitiva de los usuarios finales, con el fin de permitir que las tareas se consigan con poco trabajo y esfuerzo.
- **Simplicidad:** La construcción de la herramienta deberá considerar los diferentes niveles de conocimiento de los usuarios en el uso de aplicaciones informáticas. Es decir, considerar desde el Dueño del Producto, que usualmente posee conocimientos básicos de informática, hasta los miembros del Equipo de Desarrollo, que manejan conceptos avanzados de informática. Por lo tanto, las diferentes funcionalidades de la herramienta deben ofrecer órdenes y procesos que se puedan llevar a cabo de manera sencilla y simple.



- **Accesibilidad:** Las descripciones de las diferentes instrucciones de la herramienta deberán proveer contenidos que resulten comprensibles y claros para todos los usuarios. Además, las funcionalidades de la herramienta deberán ofrecer opciones para evitar y corregir los errores de los usuarios en su utilización.
- **Facilidad de aprendizaje:** Las funcionalidades de la herramienta deberán diseñarse con el fin de permitir que los usuarios aprendan a utilizarla sin que se requiera mucho esfuerzo o información adicional.

### 3.5.2. Diseño

Considerando los diferentes requisitos educidos, se realizó el diseño de la herramienta. En la Figura 33, se describen visualmente los diferentes componentes tecnológicos que soportan la implementación de Scrum-UIA-MAT. Como se puede observar en la Figura 33, Scrum-UIA-MAT se ha desarrollado para ejecutarse en un entorno cliente-servidor, donde las tareas se reparten entre los servidores. En este caso, el cliente corresponde a la Interfaz Web de Scrum-UIA-MAT (navegador Web), que se comunica con el servidor Web, que posee las prestaciones de *Servicios de Información de Internet* (IIS<sup>14</sup>) y *.NET Framework*. Asimismo, el servidor Web es el encargado de procesar la aplicación Scrum-UIA-MAT, realizando las conexiones (cuando se requiera) con la base de datos, y procesando las peticiones del cliente. A continuación, se describen los componentes de la infraestructura tecnológica que soportan Scrum-UIA-MAT.



**Figura 33:** Infraestructura tecnológica que soporta Scrum-UIA-MAT.

---

<sup>14</sup> Por sus siglas del inglés: **I**nternet **I**nformation **S**ervices.

### 3.5.2.1. Interfase Web de Scrum-UIA-MAT

La herramienta Scrum-UIA-MAT se ha desarrollado para un ambiente Web, es decir, el entorno Web corresponde a la interfaz gráfica que los usuarios utilizarán para acceder a la herramienta. La Interfaz Web de Scrum-UIA-MAT tendrá la implementación de todos los requisitos planteados anteriormente. Por ejemplo, esta herramienta contendrá las opciones para registrar y gestionar las tareas de desarrollo que los miembros del Equipo de Desarrollo deben definir por cada requisito seleccionado para el Sprint. Por lo tanto, dichos miembros utilizan esta Interfaz Web de Scrum-UIA-MAT con el fin de registrar o recuperar los distintos datos, relacionados con la gestión del proyecto (requisitos, tareas de desarrollo, Sprints, esfuerzo estimado de las tareas, entre otros).

La Interfaz Web de Scrum-UIA-MAT, se ha desarrollado con ASP.NET (Esposito, 2005), que es un marco de trabajo para construir aplicaciones Web dinámicas. Específicamente, se ha utilizado Visual Basic .NET para programar la Interfaz Web de Scrum-UIA-MAT, y Visual Studio 2013 como Entorno de Desarrollo Integrado (IDE<sup>15</sup>). Asimismo, para acceder a los datos y a los servicios de la base de datos de la herramienta (ver sección 3.5.2.3), se ha utilizado tanto ADO.NET como AJAX<sup>16</sup> (Garrett, 2005). Por un lado, se ha manejado ADO.NET para crear los diferentes componentes que permiten la comunicación con la base de datos. Por otro lado, se ha utilizado AJAX para mantener una comunicación asíncrona con la base de datos en un segundo plano, con el fin de efectuar cambios sobre las interfaces de la herramienta, sin necesidad de volver a procesarlas y cargarlas. Finalmente, se ha utilizado JQuery (Bibeault y Kats, 2008) para implementar funcionalidades en el cliente, basadas en JavaScript. Las tecnologías mencionadas anteriormente se utilizaron en el desarrollo de Scrum-UIA-MAT, con el fin de mejorar la interactividad, velocidad y usabilidad.

### 3.5.2.2. Servidor Web

Los códigos fuentes generados en el desarrollo de Scrum-UIA-MAT se encuentran almacenados y gestionados por el servidor Web. El servidor Web almacena todos los archivos relacionados con Scrum-UIA-MAT y se encarga de enviar respuestas a las peticiones de los clientes (navegadores Web). Como ya se mencionó en la sección 3.5.2, se utilizan los servicios de ISS como servidor Web, que corresponde a un conjunto de prestaciones para el sistema operativo de Microsoft Windows. En concreto, se han utilizado los siguientes módulos de IIS para soportar la aplicación desarrollada en ASP.NET:

- **ASP.NET:** Este módulo habilitó al servidor Web para hospedar las aplicaciones de ASP.NET.

---

<sup>15</sup> Por sus siglas del inglés: **I**ntegrated **D**evelopment **E**nvironment.

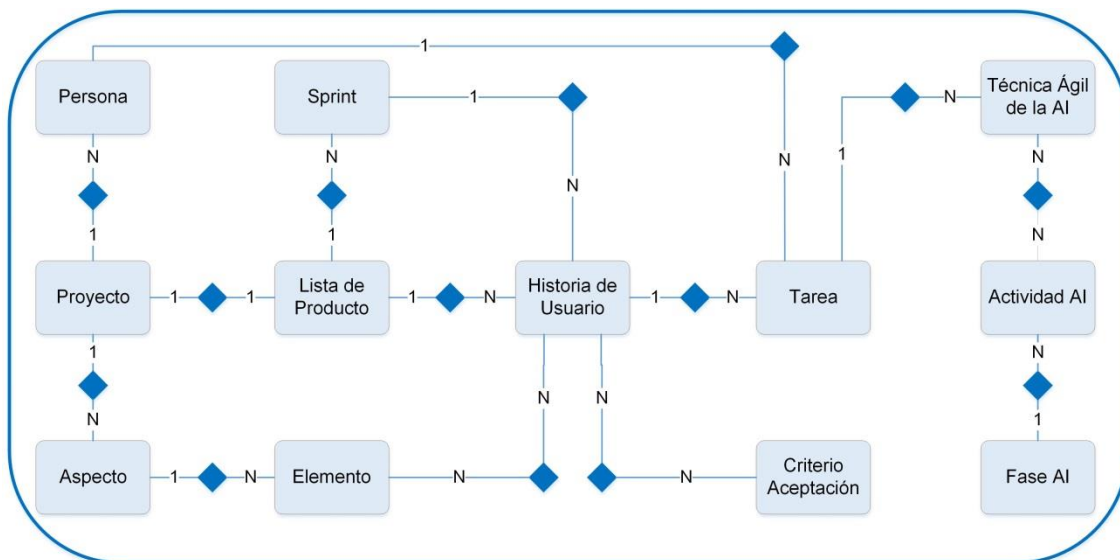
<sup>16</sup> Por sus siglas del inglés: **A**synchronous **J**avaScript **A**nd **X**ML.

- **Extensibilidad de .NET:** Este módulo permite gestionar las funcionalidades del servidor, respecto a la configuración e interfaz de usuario requeridas.
- **Filtrado de solicitudes:** Este módulo permite gestionar los tipos de solicitudes HTTP que procesa IIS, con el fin de mejorar la seguridad del servidor Web.
- **ISAPI<sup>17</sup>:** La Interfaz de Programación de Aplicaciones para Servidores de Internet (ISAPI) es una interfaz para desarrollar aplicaciones en IIS. Por lo tanto, este módulo permite alojar los archivos con código ejecutable de extensión ISAPI.
- **Extensiones ISAPI:** Este módulo permite implementar contenido Web dinámico a través de las extensiones ISAPI.

Asimismo, este servidor Web permite gestionar todas las solicitudes que se realizan en la base de datos, con el fin de registrar y recuperar los diferentes datos que manipula Scrum-UIA-MAT (usuarios finales, aspectos y elementos relevantes del proyecto, requisitos, tareas de desarrollo, criterios de evaluación de los usuarios finales, proyectos, entre otros).

### 3.5.2.3. Base de datos

Para la implementación de la base de datos de Scrum-UIA-MAT, se ha utilizado SQL Server 2008, debido a su compatibilidad con aplicaciones desarrolladas en ASP.NET y su integración en el entorno de desarrollo Visual Studio 2013. Esta base de datos corresponde a una base de datos relacional (Selinger et al., 1979), es decir, consiste en un modelo de datos, basado en la lógica de predicados y en la teoría de conjuntos.



**Figura 34:** Modelo conceptual de la base de datos.

<sup>17</sup> Por sus siglas del inglés: **I**nternet **S**erver **A**pplication **P**rogramming **I**nterface.

En la Figura 34, se presenta el modelo conceptual de la base de datos de Scrum-UIA-MAT. Este modelo conceptual permite visualizar la base de datos a través de las relaciones de sus entidades, de acuerdo a la realidad. A continuación, se describen las entidades y las relaciones principales de la base de datos.

Por un lado, las entidades reconocidas corresponden a:

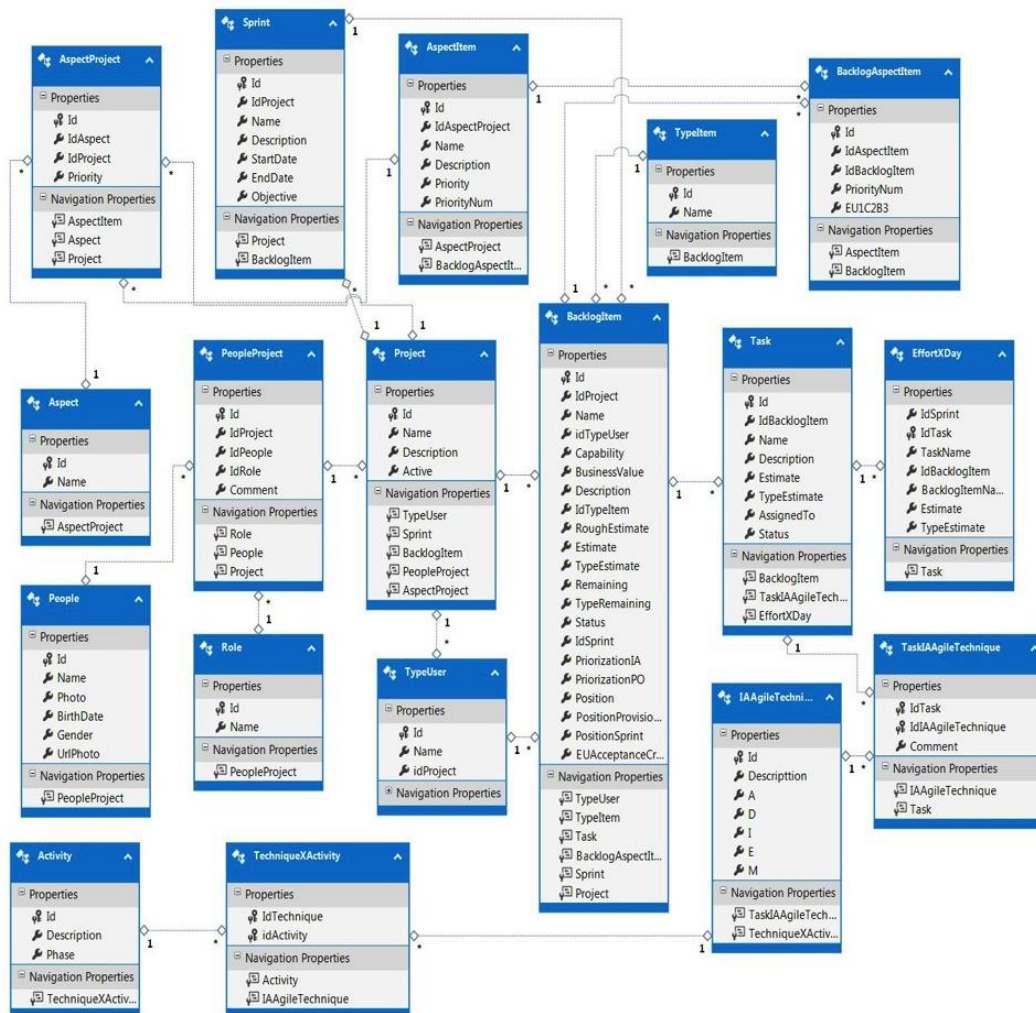
- **Proyecto:** Representa los proyectos de Scrum-UIA que se gestionan a través de la herramienta.
- **Lista del Producto:** Representa el conjunto de requisitos que se deben implementar en cada proyecto de Scrum-UIA.
- **Historias de Usuario:** Representa los requisitos que se requieren en los proyectos de Scrum-UIA.
- **Tarea:** Representa las tareas definidas por el Equipo de Desarrollo para implementar las historias de usuarios.
- **Aspecto:** Representa las características importantes, relacionadas con el proyecto.
- **Elemento:** Representa los elementos de los que se compone cada uno de los aspectos del proyecto.
- **Persona:** Representa las personas relacionadas con la gestión y ejecución del proyecto, es decir, los miembros del equipo Scrum-UIA.
- **Criterio de Aceptación:** Representa los requisitos de usabilidad de los usuarios finales.
- **Fase de la AI:** Representa las fases para integrar la AI.
- **Actividad de la AI:** Representa las actividades para integrar la AI.
- **Técnica ágil de la AI:** Representa las técnicas para integrar la AI en entornos ágiles.
- **Sprint:** Representa los Sprints que agrupan los requisitos de la Lista del Producto que se implementarán.

Por otro lado, las relaciones de este modelo corresponden a:

- Un Proyecto puede tener una Lista del Producto.
- Una Lista del Producto puede tener una o varias Historias de Usuario.
- Una Historia de Usuario puede tener una o varias Tareas.
- Una Tarea puede tener asociada una o varias Técnicas Ágiles de la AI.
- Una Técnica Ágil de la AI puede estar asociada a una o varias Actividades de la AI.
- Una Actividad de la AI puede tener asociada una o varias Técnicas Ágiles de la AI.
- Una Fase de la AI puede tener una o varias Actividades de la AI.
- Un Proyecto puede tener uno o varios Aspectos.
- Un Aspecto puede tener asociado uno o varios Elementos.
- Un Elemento puede estar asociado a una o varias Historias de Usuario.

- Una Historia de Usuario puede estar asociada a uno o varios Elementos.
- Una Historia de Usuario puede estar asociada a uno o varios Criterios de Aceptación.
- Un Criterio de Aceptación puede estar asociado a una o varias Historias de Usuario.
- Una Lista del Producto puede tener uno o varios Sprints.
- Un Sprint puede tener una o varias Historias de Usuario.
- Un Proyecto puede tener una o varias Personas.
- Una Tarea está asociada a una Persona.
- Una Persona puede tener una o varias Tareas.

Finalmente, en la Figura 35, se presenta el modelo físico de la base de datos. A diferencia del modelo conceptual, este modelo permite conocer la estructura real de la base de datos de Scrum-UIA-MAT.

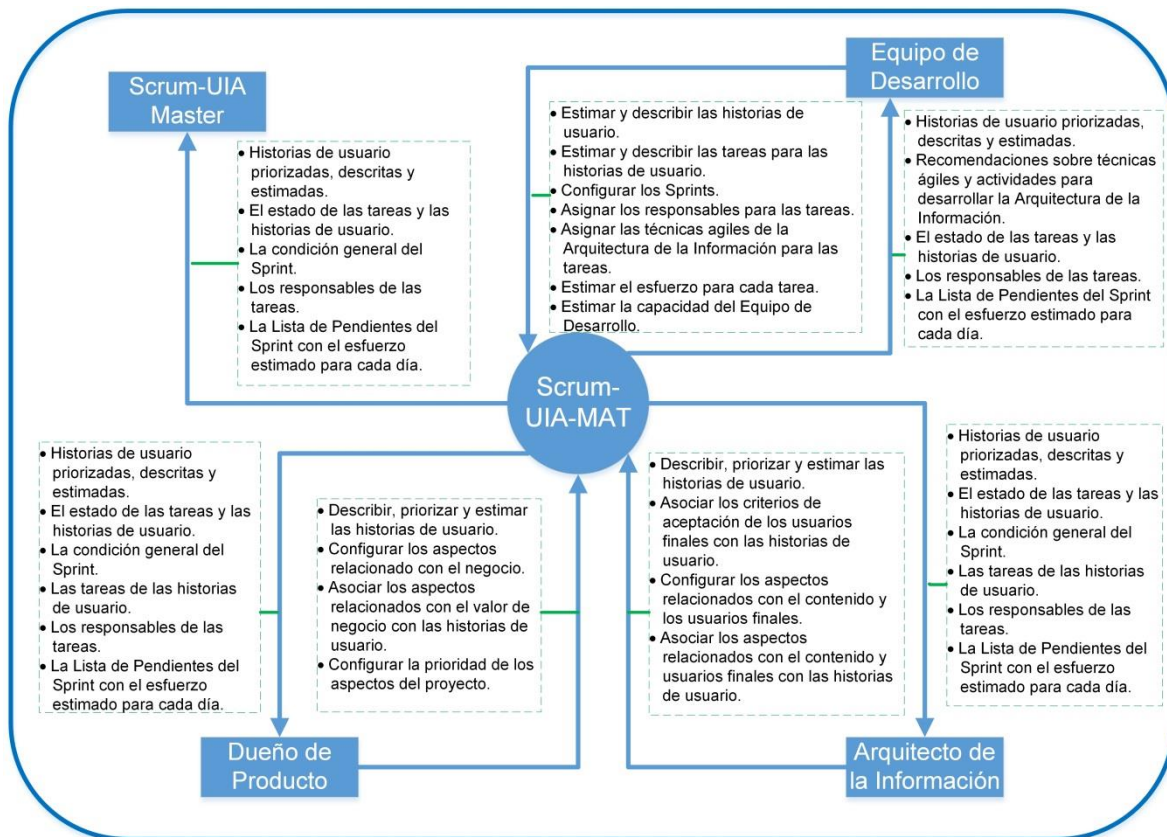


**Figura 35:** Modelo físico de la base de datos.

### 3.5.3. Descripción de Scrum-UIA-MAT

Una vez descritos los componentes tecnológicos utilizados en la implementación de la herramienta Scrum-UIA-MAT, en esta sección se describen sus funcionalidades y su aspecto final.

En primer lugar, es posible identificar las distintas entidades que interactúan con la herramienta y el flujo de información generado en cada caso. En la Figura 36, se presenta el flujo de datos, a nivel de contexto de las distintas entidades de la herramienta Scrum-UIA-MAT. Como se puede observar en la Figura 36, la herramienta Scrum-UIA-MAT tiene comunicación con cuatro entidades: Arquitecto de la Información, Dueño del Producto, Equipo de Desarrollo y Scrum-UIA Master.



**Figura 36:** Diagrama de flujo de datos – nivel de contexto.

A continuación, se describe en detalle cada una de las entidades que interactúan con la herramienta Scrum-UIA-MAT:

- **Arquitecto de la Información:** Esta entidad ingresará en la herramienta la descripción, la priorización y la estimación de las historias de usuarios, asociará los criterios de aceptación de los usuarios finales con las historias de usuarios, configurará los elementos relacionados con los aspectos de contenido y de los

usuarios finales, y asociará los elementos de los aspectos de contenido y de los usuarios finales con las historias de usuario. Por otro lado, esta entidad solicitará de la herramienta la descripción, la priorización y la estimación de las historias de usuarios, consultará el estado de las tareas y de las historias de usuarios, revisará el estado general de los Sprints, consultará las tareas descritas para las historias de usuarios y los responsables de las tareas del Equipo de desarrollo. Finalmente, revisará la Lista de Pendientes del Sprint con los esfuerzos de cada día.

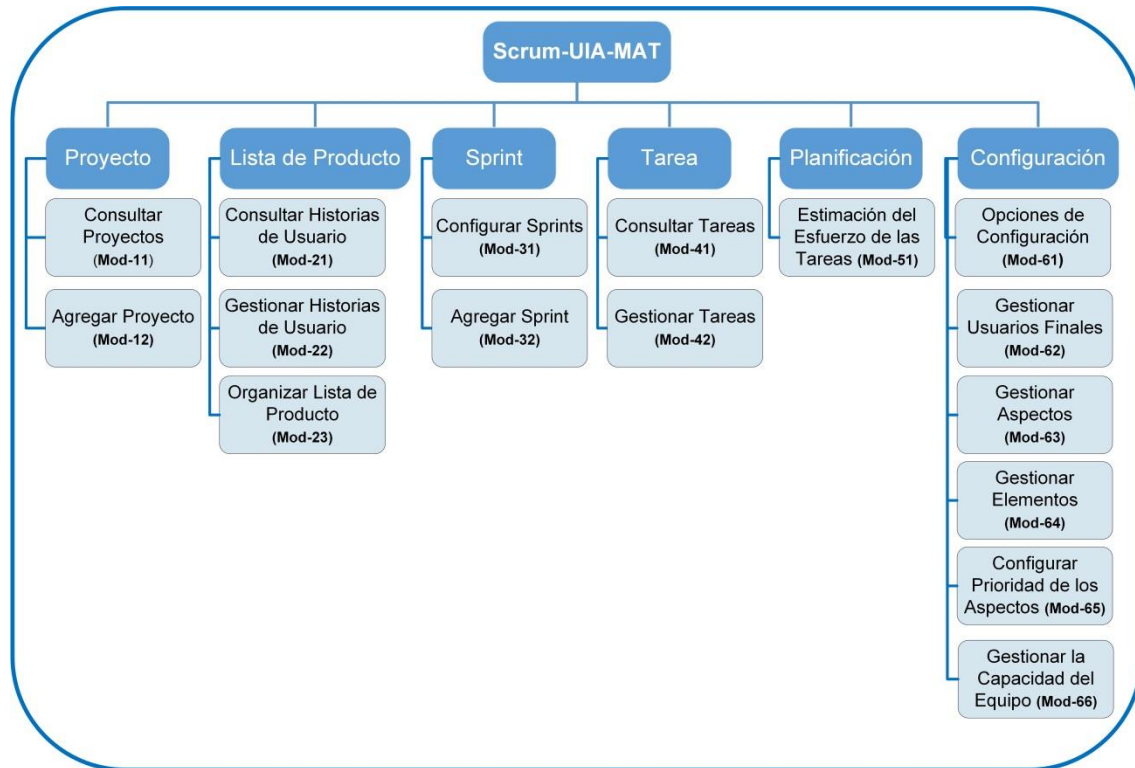
- **Equipo de Desarrollo:** Esta entidad ingresará en la herramienta la descripción y la estimación de las historias de usuarios, configurará la descripción y la estimación de tareas relacionadas con las historias de usuarios, asignará los responsables por cada una de las tareas de desarrollo, asignará en las tareas las técnicas ágiles para integrar la AI, y estimará el esfuerzo por día de las tareas. Por otro lado, esta entidad solicitará de la herramienta la descripción, la priorización y la estimación de las historias de usuarios, consultará las recomendaciones sobre técnicas ágiles y actividades para la integración de la AI, revisará el estado de las tareas y de las historias de usuarios, examinará el estado general de los Sprints y la descripción de las tareas de las historias de usuarios, revisará los responsables por cada una de las tareas del Equipo de desarrollo y la Lista de Pendientes del Sprint con los esfuerzos de cada día.
- **Dueño del Producto:** Esta entidad ingresará en la herramienta la descripción, la priorización y la estimación de las historias de usuarios, configurará todos los elementos relacionados con el aspecto de valor de negocio, asociará los elementos del aspecto de valor de negocio con las historias de usuarios, y configurará las prioridades entre los aspectos del proyecto. Por otro lado, esta entidad solicitará de la herramienta la descripción, la priorización y la estimación de las historias de usuarios, revisará el estado de las tareas y de las historias de usuarios, consultará el estado general de los Sprints, examinará las tareas descritas para las historias de usuarios y los responsables de las tareas del Equipo de desarrollo. Finalmente, revisará la Lista de Pendientes del Sprint con los esfuerzos de cada día.
- **Scrum-UIA Master:** Esta entidad solicitará de la herramienta la descripción, la priorización y la estimación de las historias de usuarios, consultará las recomendaciones sobre las técnicas ágiles y las actividades para integrar la AI, revisará el estado de las tareas y de las historias de usuarios, examinará el estado general de los Sprints, revisará la descripción de las tareas de las historias de usuarios, consultará los responsables de las tareas del Equipo de desarrollo y la Lista de Pendientes del Sprint con los esfuerzos de cada día.

Por otro lado, la herramienta Scrum-UIA-MAT tiene una navegación distribuida en diferentes ramas, con el fin de agrupar sus funcionalidades. En la Figura 37, se presenta el plano del sitio Web de Scrum-UIA-MAT. Como se puede observar en la Figura 37, la



navegación de Scrum-UIA-MAT se divide en seis ramas: Proyecto, Lista del Producto, Sprint, Tarea, Planificación y Configuración. Además, en cada una de las ramas se identifican los módulos que contienen.

La herramienta Scrum-UIA-MAT tiene un total de 16 módulos que tienen como objetivo soportar las tareas y actividades de Scrum-UIA, que generan flujos de datos finales que se requieren manipular (sistematizados) para completar correctamente el proceso de Scrum-UIA.



**Figura 37:** Plano del sitio Web de Scrum-UIA-MAT.

Asimismo, es posible identificar los tipos de acceso de los usuarios en los módulos de Scrum-UIA-MAT. En la Tabla 24, se identifican los diferentes perfiles y el tipo de permiso que éstos tienen en cada módulo de Scrum-UIA-MAT. Como se muestra en la Tabla 24, Scrum-UIA-MAT define cuatro perfiles: Arquitecto de la Información, Dueño del Producto, Equipo de Desarrollo y Scrum-UIA Master. Estos perfiles se han identificado mediante el diagrama de flujo de datos (nivel contexto) de la herramienta (ver Figura 36). Además, se definen tres tipos de permisos que pueden tener los perfiles para acceder a los módulos: *Lectura*, *Escritura* y *Responsable*.

El tipo de permiso *Lectura* identifica que el perfil solamente tiene acceso a consultar los datos que se gestionan en el módulo. El tipo de permiso *Escritura* identifica que el perfil tiene acceso para insertar, editar y consultar los datos que se gestionan en el módulo.



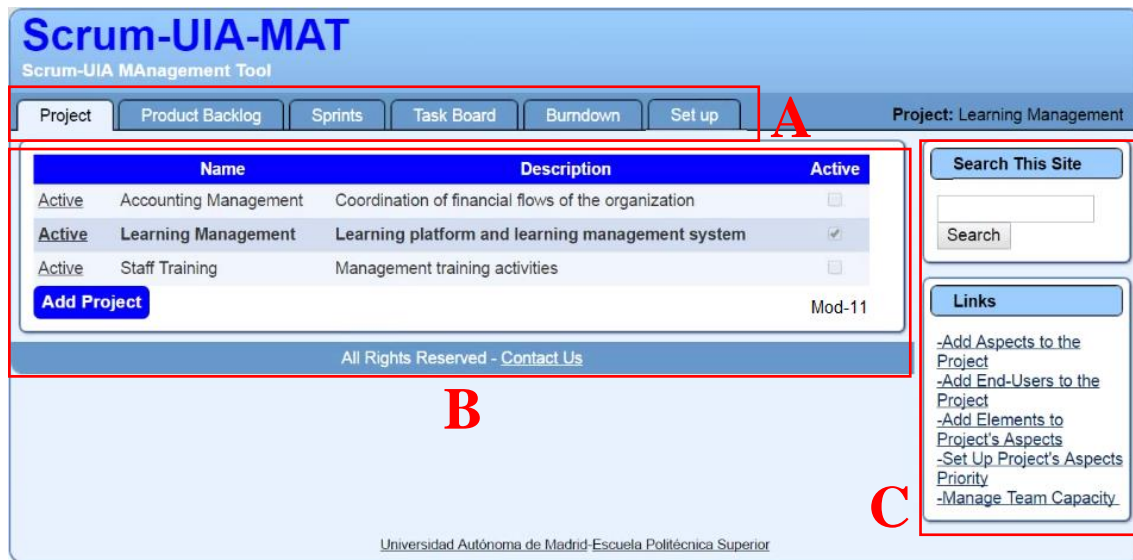
Finalmente, el tipo de permiso *Responsable* indica que el perfil tiene acceso a *Escritura*, y que es el responsable del resultado final de los datos que se gestionan en el módulo. Por ejemplo, tal y como se muestra en la Tabla 24, en el módulo “Mod-22”, que permite gestionar (describir, estimar y priorizar) las historias de usuarios, el Arquitecto de la Información y el Equipo de Desarrollo tienen acceso a *Escritura*, y el Scrum-UIA Master tiene acceso a *Lectura*. No obstante, el Dueño del Producto es el responsable final de cada una de las historias de usuario. Finalmente, en los casos donde los perfiles no tienen acceso a los módulos, se identifican con la etiqueta (-).

Módulos	Perfiles			
	Arquitecto de la Información	Dueño del Producto	Equipo de Desarrollo	Scrum-UIA Master
Mod-11	Responsable	Lectura	Lectura	Lectura
Mod-12	Responsable	Escritura	-	-
Mod-21	Lectura	Responsable	Lectura	Lectura
Mod-22	Escritura	Responsable	Escritura	Lectura
Mod-23	Lectura	Responsable	Lectura	Lectura
Mod-31	Lectura	Lectura	Responsable	Lectura
Mod-32	Lectura	Lectura	Responsable	-
Mod-41	Lectura	Lectura	Responsable	Lectura
Mod-42	Lectura	Lectura	Responsable	Lectura
Mod-51	Lectura	Lectura	Responsable	Lectura
Mod-61	Responsable	Lectura	Lectura	Lectura
Mod-62	Responsable	Escritura	Lectura	Lectura
Mod-63	Responsable	Escritura	Lectura	Lectura
Mod-64	Responsable	Escritura	Lectura	Lectura
Mod-65	Responsable	Escritura	Lectura	Lectura
Mod-66	Lectura	Lectura	Responsable	Lectura

**Tabla 24:** Identificación de los perfiles en Scrum-UIA-MAT y el tipo de permiso que éstos tienen en cada módulo.

Finalmente, en la Figura 38, se presenta la estructura general de Scrum-UIA-MAT, a la que pueden acceder los usuarios una vez autenticados. La autenticación de los usuarios se

realiza a través de un formulario, donde se debe ingresar el nombre del usuario y su contraseña. En la parte A de la Figura 38, se muestra el encabezado de la herramienta, correspondiente a las seis ramas de navegación (identificadas en la Figura 37). En la parte B de la Figura 38, se despliegan los diferentes módulos de Scrum-UIA-MAT. Por último, en la parte C de la Figura 38, se listan los enlaces de acceso directo para los módulos de Scrum-UIA-MAT.



**Figura 38:** Interfaz de Scrum-UIA-MAT, dividida en tres partes: A, B y C.

A continuación, se presenta y describe en detalle cada uno de los módulos de Scrum-UIA-MAT, agrupados por las diferentes ramas de navegación, es decir, Proyecto (Project), Lista del Producto (Product Backlog), Sprints, Tareas (Task Board), Planificación (Burndown) y Configuración (Set Up):

#### 3.5.3.1. Proyecto (Project)

En esta opción de Scrum-UIA-MAT, se agrupan los módulos que permiten consultar y manipular los proyectos software que son gestionados con la herramienta Scrum-UIA-MAT. Esta opción tiene dos módulos:

##### a) Consultar Proyectos (Mod-11)

En la parte B de la Figura 38, se puede observar el módulo Mod-11. Éste permite consultar los proyectos software que son gestionados con Scrum-UIA-MAT y activar (opción Active) el proyecto que se pretende configurar. Asimismo, se provee la opción (botón Add Project) para acceder al módulo para agregar nuevos proyectos software (Mod-12). El Arquitecto de la Información es el *Responsable* de este módulo, mientras que el Dueño del Producto, el Equipo de Desarrollo y el Scrum-UIA Master tienen acceso a *Lectura*. Finalmente, este módulo sólo tiene la siguiente salida:

### Salida

- Los proyectos software, gestionados con Scrum-UIA-MAT.

#### b) Agregar Proyecto (Mod-12)

El módulo Mod-12 (ver Figura 39) permite añadir nuevos proyectos en Scrum-UIA-MAT, al que se accede a través del módulo “Consultar Proyectos (Mod-11)” (botón Add Project). Los campos requeridos corresponden al nombre y la descripción del proyecto (campos Name y Description, respectivamente). Una vez guardado el formulario (botón Add Project), se carga el módulo “Consultar Proyectos (Mod-11)” con los datos actualizados. Solamente el Arquitecto de la Información y el Dueño del Producto tienen acceso a este módulo, en términos de *Responsable* y *Escritura*, respectivamente.

### Entradas

- El nombre del proyecto.
- La descripción del proyecto.

### Salida

- La determinación de los proyectos.



**Figura 39:** Módulo para ingresar proyectos (Mod-12).

#### 3.5.3.2. Lista del Producto (Product Backlog)

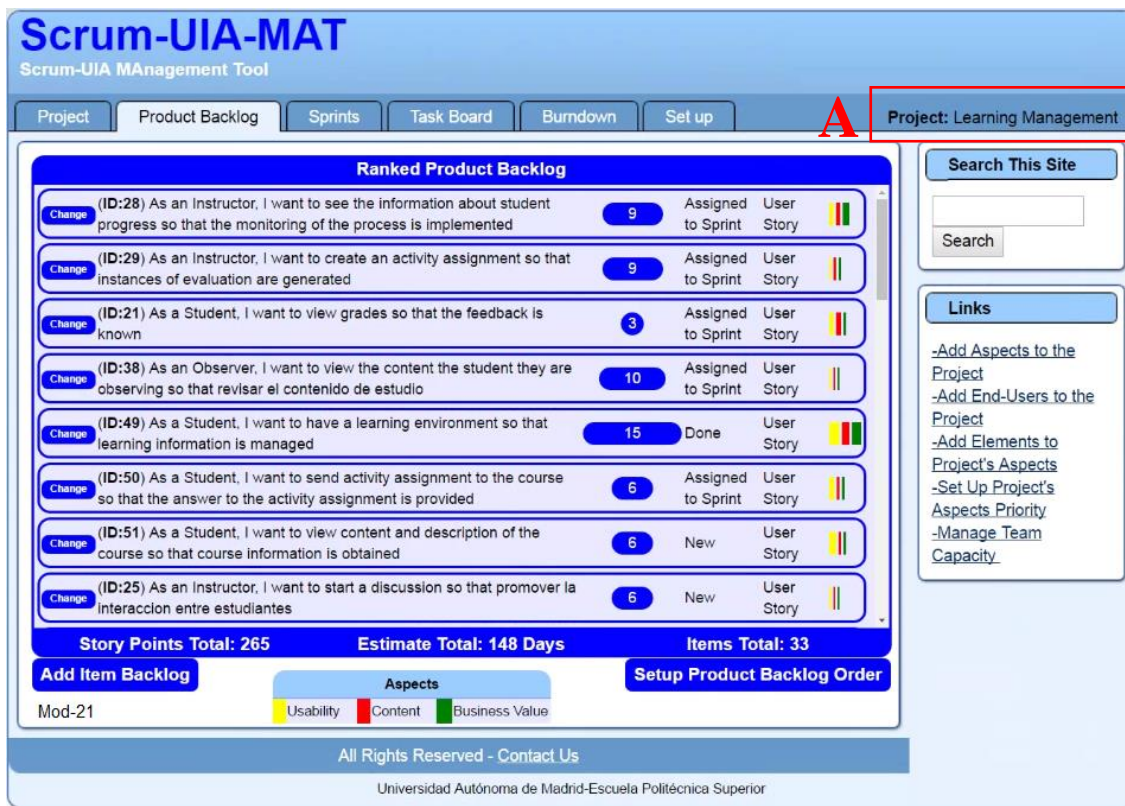
Esta opción agrupa a los módulos que permiten gestionar (consultar, insertar, modificar, eliminar y organizar) los requisitos de cada proyecto. La opción Lista del Producto contiene tres módulos:

##### a) Consultar Historias de usuarios (Mod-21)

En la Figura 40, se presenta el módulo que permite consultar las historias de usuarios (definidas en la “Actividad 2.2. Generación final de requisitos” de Scrum-UIA). Este módulo lista todas las historias de usuarios, asociadas al proyecto seleccionado y activado en el módulo “Consultar Proyectos (Mod-11)” (en la parte A de la Figura 40, se puede visualizar el proyecto activado). Como se muestra en la Figura 40, las historias de usuarios son listadas con datos referentes a su código de identificación, nombre, estimación, estado,

tipo de historia de usuario y una identificación de los aspectos relevantes asociados (colores al final de la descripción).

Cabe destacar que los diferentes colores (amarillo, rojo y verde), que se indican al final de la descripción de cada requisito (ver Figura 40), se utilizan para indicar, de manera gráfica, el nivel de relevancia que tienen los requisitos en cada aspecto (Usability, Content y Business Value). Este nivel de relevancia se obtiene como resultado del proceso de priorización (ver a continuación el Mod-22), mediante el enfoque del método de priorización propuesto (QMPSR) para Scrum-UIA. En el caso que se muestra en la Figura 40, los colores amarillo, rojo y verde indican el *Nivel de Relevancia de los Requisitos por Aspecto* ( $\lambda$ ) para los aspectos Usability, Content y Business Value, respectivamente, tal y como se define en la fórmula (8). Para mayor detalle, revisar la sección 3.3.



**Figura 40:** Módulo para consultar historias de usuarios (Mod-21). Los colores amarillo, rojo y verde indican el nivel de relevancia que tienen los requisitos en cada aspecto relevante del proyecto, de acuerdo al enfoque del método de priorización QMPSR.

Asimismo, en la parte inferior del módulo se encuentran las opciones Add Item Backlog y Setup Product Backlog Order, que permiten acceder a los módulos para gestionar las historias de usuarios (Mod-22) y organizar la Lista del Producto (Mod-23), respectivamente. Finalmente, el Dueño del Producto es el *Responsable* de este módulo,

mientras que el Arquitecto de la Información, el Equipo de Desarrollo y el Scrum-UIA Master tienen acceso a *Lectura*. La entrada y la salida de este módulo corresponden a:

**Entrada**

- El proyecto seleccionado y activado en el módulo “Consultar Proyectos (Mod-11)”.

**Salida**

- La Lista del Producto con los requisitos descritos en formato de historias de usuarios.

**b) Gestionar Historias de usuarios (Mod-22)**

La Figura 41 muestra el módulo para gestionar las historias de usuario. Este módulo permite describir, estimar y priorizar los requisitos de cada proyecto (realizado en la Tarea 2.2.2, la Tarea 2.2.3 y la Tarea 2.2.4 de Scrum-UIA, respectivamente). Se accede a este módulo a través del botón *Add Item Backlog* del módulo “Consultar Historias de usuarios (Mod-21)” (ver Figura 40). Como se puede observar en la Figura 41, se ha etiquetado con letras (A, B, C, D y E) para señalar los diferentes apartados de este módulo. De este modo, en la etiqueta A, se encuentra el apartado que permite describir los requisitos en formato de historias de usuarios e indicar los datos relacionados con la estimación y el estado actual de la historia de usuario. En las etiquetas B y C, se encuentran las dos opciones para priorizar las historias de usuario. Por un lado, en la etiqueta B, se halla la opción que permite al Dueño de Proyecto y al Arquitecto de la Información priorizar las historias de usuarios de manera independiente, es decir, los requisitos se priorizan de acuerdo a sus preferencias de manera separada. Por otro lado, en la etiqueta C, se encuentra la segunda opción para priorizar las historias de usuario. Esta funcionalidad permite priorizar los requisitos mediante los elementos y los aspectos relevantes del proyecto, es decir, a través del enfoque de priorización propuesto anteriormente en QMPSR (ver sección 3.3). Asimismo, en la etiqueta D, se encuentra la funcionalidad para identificar las dependencias de las historias de usuario. Por último, en la etiqueta E, está el apartado que permite señalar los criterios de aceptación de los usuarios finales de cada historia de usuario.

Una vez guardado el formulario (botón *Add Item*), se carga el módulo “Consultar Historias de Usuarios (Mod-21)” con las historias de usuarios actualizadas. Finalmente, el Dueño del Producto es el *Responsable* de este módulo, el Scrum-UIA Master sólo tiene acceso a *Lectura*, y el Arquitecto de la Información y el Equipo de Desarrollo tienen acceso a *Escritura*. Las entradas y la salida de este módulo corresponden a:

**Entradas**

- El proyecto seleccionado y activado en el módulo “Consultar Proyectos (Mod-11)”.

- Los usuarios finales definidos en el módulo “Gestionar Usuarios Finales (Mod-62)”.
- La descripción y la estimación de la historia de usuario.
- Los Sprints definidos en el módulo “Agregar Sprint (Mod-32)”.
- La prioridad del Dueño del Producto y el Arquitecto de la Información.
- Los aspectos relevantes del proyecto, definidos en el módulo “Gestionar Aspectos (Mod-63)”.
- Los elementos de los aspectos relevantes del proyecto, definidos en el módulo “Gestionar Elementos (Mod-64)”.
- Las historias de usuarios definidas anteriormente mediante este módulo.
- Los criterios de aceptación de los usuarios finales.

### Salida

- La historia de usuario descrita, estimada y priorizada.

The screenshot displays the 'Backlog Item' form in Mod-22. The form is divided into several sections, each with a red letter label (A through E) on the left side. Section A contains the 'Project' dropdown (set to 'Learning Management'), 'Name' field (filled with 'As a Student, I want to view events and so that course information'), 'Description' text area (filled with 'Students should view events and assignments in their courses and groups, in order to obtain course information'), 'Rough Estimate' (6 story points), 'Type' dropdown (User Story), 'Estimate' (2 Days), 'Remaining' (2 Days), 'Status' dropdown (New), and 'Sprint' dropdown (Select Sprint). Section B, labeled 'Priority', contains 'Product Owner' and 'Information Architect' buttons, both set to 'Set Up'. Section C, labeled 'Related Aspects', contains 'Usability' (Provide a simple design), 'Contents' (Course activities), and 'Business Value' (Management of the entities of knowledge). Section D, labeled 'Dependencies', contains 'User Story' (2 selected). Section E, labeled 'End-User Acceptance Criteria', contains a text area with the text 'The presentation of information should provide adaptable alternatives'. At the bottom of the form, there is a 'Mod-22' label and two buttons: 'Add Item' and 'Cancel'.

**Figura 41:** Módulo para gestionar historias de usuarios (Mod-22).

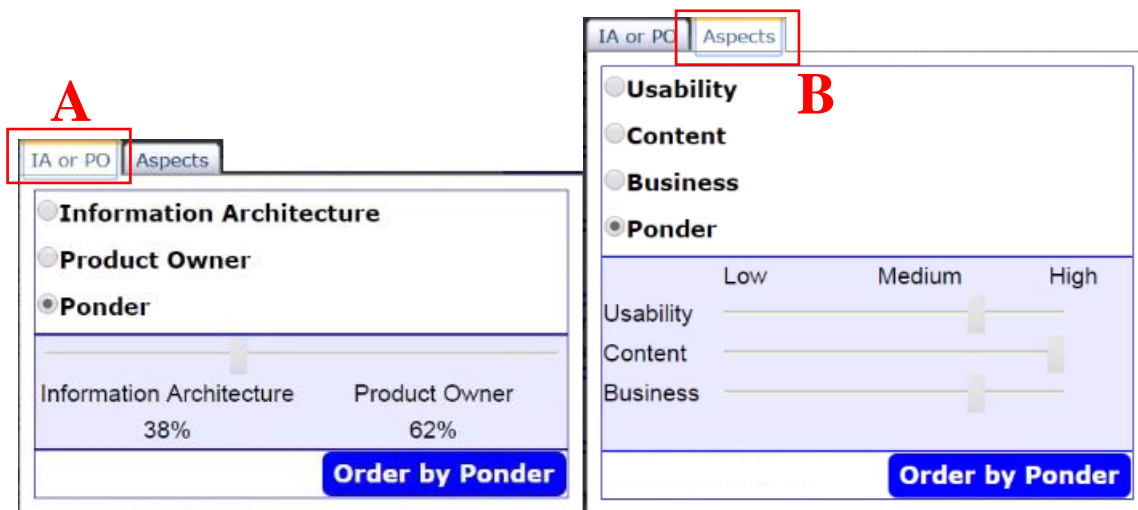


### c) Organizar Lista del Producto (Mod-23)

Este módulo (ver Figura 42) permite ordenar la Lista del Producto, de acuerdo a las dos opciones ofrecidas para priorizar los requisitos (ver las etiquetas B y C del módulo “Gestionar Historias de usuarios (Mod-22)”). Se accede a este módulo a través del botón `Setup Product Backlog Order` del módulo “Consultar Historias de usuarios (Mod-21)”.

Por un lado, en la parte izquierda de la Figura 42, se muestra la opción para ordenar la Lista del Producto, de acuerdo a las prioridades que fueron entregadas por el Dueño del Producto y el Arquitecto de la Información de forma independiente (definidas en la Tarea 2.2.4 de Scrum-UIA). En la parte A de la Figura 42, se señala cómo acceder a esta opción del módulo. Allí se puede observar que los requisitos se pueden ordenar de acuerdo a las prioridades entregadas por el Dueño del Producto o el Arquitecto de la Información, o a través de una ponderación, en términos porcentuales, de las prioridades entregadas por ellos.

Por otro lado, en la parte derecha de la Figura 42, se señala la segunda opción para ordenar la Lista del Producto, que se realiza de acuerdo a las prioridades de los elementos y de los aspectos relevantes del proyecto (definidos también en la Tarea 2.2.4 de Scrum-UIA). Asimismo, en la parte B de la Figura 42, se señala cómo acceder a esta opción del módulo. Como se puede observar, los requisitos se pueden ordenar de acuerdo a alguno de los aspectos relevantes, definidos en el proyecto (en este caso, *Usability*, *Content* y *Business* corresponden a los aspectos relevantes del proyecto, que son definidos en el módulo “Gestionar Aspectos (Mod-63)”) o a través de una ponderación de las prioridades de cada uno de los aspectos relevantes del proyecto. Estas funcionalidades del módulo corresponden a la implementación del enfoque de priorización propuesto en la sección 3.3.



**Figura 42:** Módulo para organizar Lista de Proyecto (Mod-23).

Una vez configurados y establecidos los parámetros para ordenar la Lista del Producto (botón `Order by Ponder`), se carga el módulo “Consultar Historias de usuarios (Mod-21)” con las historias de usuarios ordenadas de acuerdo a los criterios indicados. Finalmente, el Dueño del Producto es el *Responsable* de este módulo, y el Arquitecto de la Información, el Equipo de Desarrollo y el Scrum-UIA Master tienen acceso a *Lectura*. Las entradas y la salida de este módulo corresponden a:

#### **Entradas**

- La prioridad del Dueño del Producto y el Arquitecto de la Información.
- Los aspectos relevantes del proyecto, definidos en el módulo “Gestionar Aspectos (Mod-63)”.

#### **Salida**

- La Lista del Producto, ordenada de acuerdo a las prioridades del Dueño del Producto y/o el Arquitecto de la Información, o los aspectos relevantes del proyecto.

#### **3.5.3.3. Sprint**

Los módulos para configurar cada Sprint del proyecto se agrupan en esta opción. La opción Sprint contiene dos módulos:

##### **a) Configurar Sprints (Mod-31)**

Este módulo (ver Figura 43) permite seleccionar y organizar las historias de usuarios que se desarrollan en cada Sprint (correspondiente a la “Actividad 3.3. Selección de requisitos para el Sprint” de Scrum-UIA). En la Figura 43, se muestran las historias de usuarios (ver apartado `Ranked Producto Backlog` de la Figura 43), ordenadas de acuerdo a la configuración establecida en el módulo “Organizar Lista del Producto (Mod-23)”. Asimismo, en la parte izquierda de este módulo, se encuentran los requisitos seleccionados para el Sprint 1 (ver campo `Sprint`) del proyecto `Learning Management` (ver parte A de la Figura 43), que se pueden reorganizar para su implementación en el Sprint, de acuerdo a las preferencias del Equipo de Desarrollo.

Además, en la parte inferior izquierda de este módulo, se tiene la opción (botón `Add Sprint`) para acceder al módulo que permite agregar Sprints al proyecto (Mod-32). Finalmente, el Equipo de Desarrollo es el *Responsable* de este módulo, mientras que el Arquitecto de la Información, el Dueño del Producto y el Scrum-UIA Master tienen acceso a *Lectura*. A continuación, se describen las entradas y las salidas:

#### **Entradas**

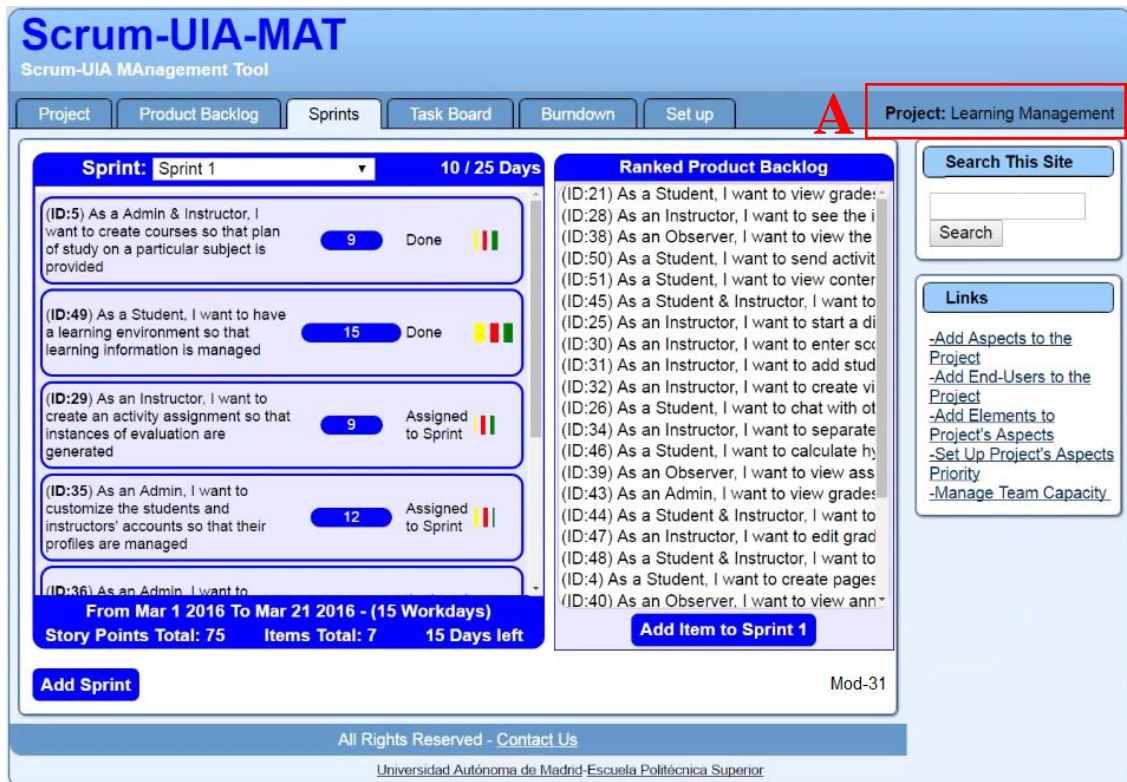
- El proyecto seleccionado y activado en el módulo “Consultar Proyectos (Mod-11)”.
- Los Sprints definidos en el módulo “Agregar Sprint (Mod-32)”.



- Las historias de usuarios definidas en el módulo “Gestionar Historias de usuarios (Mod-22)” y ordenadas de acuerdo a la configuración establecida en el módulo “Organizar Lista del Producto (Mod-23)”.
- Las historias de usuarios seleccionadas para cada Sprint.

### Salidas

- Las historias de usuarios para cada Sprint del proyecto.
- Los datos de configuración de cada Sprint del proyecto.



**Figura 43:** Módulo para configurar Sprints (Mod-31).

#### b) Agregar Sprint (Mod-32)

Este módulo (ver Figura 44) permite especificar los datos necesarios para configurar cada Sprint del proyecto. Como se muestra en la Figura 44, se proveen opciones para detallar la duración (ver campos *Start Date* y *End Date*), el objetivo (ver campo *Objective*) y la definición de “Terminado” (ver campo *Definition of Done*) de cada Sprint (correspondientes a la Tarea 3.1.1, la Tarea 3.2.1<sup>18</sup> y la Tarea 2.3.1 de Scrum-UIA, respectivamente). Una vez guardado el formulario (botón *Add Sprint*), se carga el módulo “Configurar Sprints (Mod-31)” con los datos del Sprint actualizados. Finalmente,

<sup>18</sup> Este módulo también soporta la identificación del objetivo, inicialmente propuesto en la Tarea 2.4.1.

el Equipo de Desarrollo es el *Responsable* de este módulo, mientras que el Arquitecto de la Información y el Dueño del Producto sólo tienen acceso a *Lectura*. Las entradas y la salida de este módulo corresponden a:

#### Entradas

- El proyecto seleccionado y activado en el módulo “Consultar Proyectos (Mod-11)”.
- La descripción del Sprint.
- Las fechas de inicio y fin del Sprint.
- El objetivo del Sprint.
- La definición de “Terminado del Sprint”.

#### Salida

- El Sprint descrito y planificado.

**Sprint**

Project:

Sprint Name:

Description:

Objective:

Definition of Done:

Start Date:  End Date:

Mod-32

**Figura 44:** Módulo para agregar Sprint (Mod-32).

#### 3.5.3.4. Tareas (Task Board)

Esta opción contiene los módulos para gestionar (agregar, modificar, eliminar y consultar) las tareas que se realizan para implementar las historias de usuarios, seleccionadas en cada Sprint. A continuación, se describen los módulos que contiene esta opción:

#### a) Consultar Tareas (Mod-41)

Este módulo permite consultar las tareas de desarrollo, agrupadas por historia de usuario y Sprint. Es decir, por cada Sprint se listan las historias de usuarios contenidas, y por cada historia de usuario se indican las tareas de desarrollo definidas para llevar a cabo. Tal y como se puede observar, en la Figura 45 se presenta este módulo: en la parte inferior izquierda, se encuentra la opción (botón Manage Tasks) para acceder al módulo que permite configurar cada tarea (Mod-42). Finalmente, el Equipo de Desarrollo es el *Responsable* de este módulo, mientras que el Arquitecto de la Información, el Dueño del Producto y el Scrum-UIA Master tienen acceso a *Lectura*.

**Scrum-UIA-MAT**  
Scrum-UIA MAnagement Tool

Project | Product Backlog | Sprints | Task Board | Burndown | Set up | Project: Learning Management

Sprint: Sprint 1 | From Mar 1 2016 To Mar 21 2016 - (15 Workdays)

Backlog Item	Status	Type	Total		
(ID:5) As a Admin & Instructor, I want to create courses so that plan of study on a particular subject is provided	Done	User Story	5 Tasks, 40 Hours		
<b>Id</b>	<b>Name</b>	<b>Estimate</b>	<b>Status</b>	<b>IA Techniques</b>	<b>Assigned To</b>
3	Create algorithm to record student's activity in a course	8 Hours	Done	Card Sorting, Consistency inspection	Development Team
20	Content model courses	4 Hours	Done	Card Sorting, Interviews*	Information Architect
21	Class diagram courses	4 Hours	Done		Development Team
22	Wireframe to create courses	8 Hours	Done	Lo-fi prototyping*	Information Architect
23	Programming algorithm to create courses	16 Hours	Done		Development Team
(ID:49) As a Student, I want to have a learning environment so that learning information is managed	Done	User Story	5 Tasks, 38 Hours		
(ID:29) As an Instructor, I want to create an activity assignment so that instances of evaluation are generated	Assigned to Sprint	User Story	4 Tasks, 24 Hours		
(ID:35) As an Admin, I want to customize the students and instructors' accounts so that their profiles are managed	Assigned to Sprint	User Story	4 Tasks, 30 Hours		
(ID:36) As an Admin, I want to manage course enrollments so that the course composition is managed	Assigned to Sprint	User Story	4 Tasks, 30 Hours		
(ID:22) As a Student, I want to view events and assignments of courses so that course information is obtained	Assigned to Sprint	User Story	1 Tasks, 12 Hours		
(ID:16) As an Instructor, I want to add course content so that the material of the course is provided	In progress	User Story	4 Tasks, 30 Hours		

**Manage Tasks** | Mod-41

All Rights Reserved - Contact Us  
Universidad Autónoma de Madrid-Escuela Politécnica Superior

**Search This Site**  
Search

**Links**  
-Add Aspects to the Project  
-Add End-Users to the Project  
-Add Elements to Project's Aspects  
-Set Up Project's Aspects Priority  
-Manage Team Capacity

Figura 45: Módulo para consultar tareas de desarrollo (Mod-41).

A continuación, se describen las entradas y la salida de este módulo:

#### Entradas

- El proyecto seleccionado y activado en el módulo “Consultar Proyectos (Mod-11)”.
- Los Sprints definidos en el módulo “Agregar Sprint (Mod-32)”.
- Las historias de usuarios seleccionadas para cada Sprint, definidas en el módulo “Configurar Sprint (Mod-31)”.
- Las tareas de desarrollo definidas en cada historia de usuario.

## Salida

- Las tareas de desarrollo agrupadas por las historias de usuarios y Sprint.

### b) Gestionar Tareas (Mod-42)

En la Figura 46, se presenta el módulo para gestionar las tareas de desarrollo. Este módulo permite determinar las tareas de desarrollo para los requisitos seleccionados del Sprint (Tarea 3.4.1 de Scrum-UIA), identificar las técnicas ágiles de la AI para las tareas (Tarea 3.4.2 de Scrum-UIA), estimar su duración (Tarea 3.4.3 de Scrum-UIA) y asignar los responsables de las tareas (Tarea 3.4.4 de Scrum-UIA).

La configuración de las tareas de desarrollo se especifica mediante los siguientes campos: el nombre, la descripción, la estimación, el estado, el responsable y el estado de la tarea (correspondientes a los campos *Name*, *Description*, *Estimate*, *Status* y *Assigned to*, respectivamente). Asimismo, la configuración de las técnicas ágiles de la configuración se encuentra en el apartado *Incorporate Information Architecture Agile Techniques*, que provee el conjunto de técnicas, obtenidas como resultado del análisis a las nueve propuestas de la AI (Rocio y Cerutti, 2016; Rosenfeld et al., 2015; Zhang et al., 2015; Nurcahyanti, 2014; Picchi, 2011; Sharlin et al., 2009; Reichenauer, 2005; Lamar, 2001; Toub, 2000). Este conjunto también incluye la técnica propuesta (InterArch-T) para fomentar el desarrollo incremental de los requisitos en Scrum-UIA.

Al guardar el formulario (botón *Accept*), se carga el módulo “Consultar Tareas (Mod-41)” con las tareas de desarrollo actualizadas. Finalmente, el Equipo de Desarrollo es el *Responsable* de este módulo, y el Arquitecto de la Información, el Scrum-UIA Master y el Dueño del Producto sólo tienen acceso a *Lectura*. Las entradas y la salida de este módulo corresponden a:

## Entradas

- El proyecto seleccionado y activado en el módulo “Consultar Proyectos (Mod-11)”.
- El Sprint seleccionado en el módulo “Consultar Tareas (Mod-41)”.
- Las historias de usuarios, escogidas para el Sprint seleccionado en el módulo “Consultar Tareas (Mod-41)”.
- Las tareas de desarrollo, escogidas para el Sprint seleccionado en el módulo “Consultar Tareas (Mod-41)”.
- El nombre de la tarea de desarrollo.
- La descripción de la tarea de desarrollo.
- La estimación de la tarea de desarrollo.

- Los responsables definidos en el módulo “Gestionar la Capacidad del Equipo (Mod-66)”.
- El estado de la tarea de desarrollo.
- Las técnicas ágiles de la AI, agrupadas por las fases, y las actividades que desarrolla la AI.

### Salida

- Las tareas de desarrollo, descritas y estimadas.

The screenshot displays the 'Manage Tasks' interface. At the top, the title 'Manage Tasks' is in a blue bar. Below it, the 'Backlog Item' is '(ID:22) As a Student, I want to view events and assignments of courses so t'. The 'Task' dropdown is set to 'New task'. The 'Name' field contains 'Content model courses'. The 'Description' field contains 'Identify all content entities that compose a course.'. The 'Estimate' is '6' with a unit of 'Days'. The 'Assigned to' dropdown is 'Information Architect'. The 'Status' dropdown is 'To do'. Below this, a section titled 'Incorporate Information Architecture Agile Techniques' contains a 'Filter Techniques' button and a list of techniques: 'Interface design patterns', 'Lo-fi prototyping\*', 'Mock-up prototype', and 'Storyboards'. There are 'Add' and 'Remove' buttons. Below the list, the 'Added IA Techniques' section shows 'InterArch-T' and 'Diagramming'. At the bottom left is 'Mod-42' and at the bottom right are 'Accept' and 'Cancel' buttons.

**Figura 46:** Módulo para gestionar las tareas de desarrollo (Mod-42).

### 3.5.3.5. Planificación (Burndown)

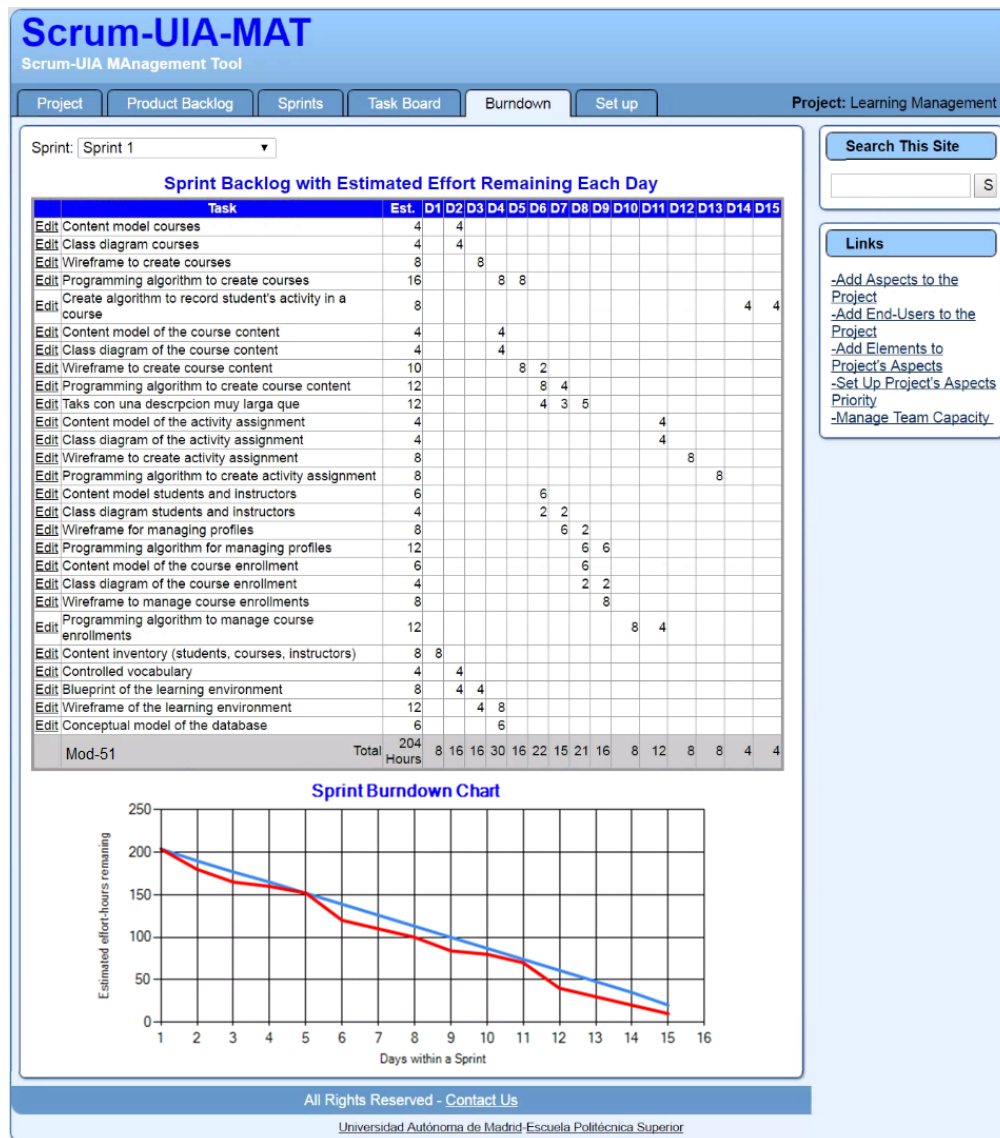
Esta opción sólo contiene un módulo, que permite planificar las tareas de desarrollo, señalando su esfuerzo diario durante el Sprint. A continuación, se presenta el módulo que contiene esta opción:

#### a) Estimación del Esfuerzo de las Tareas (Mod-51)

En la Figura 47, se presenta el módulo para estimar el esfuerzo diario de las tareas de desarrollo (correspondiente a la “Actividad 3.5. Planificación de las tareas de desarrollo”



de Scrum-UIA). En este módulo, se listan las tareas de desarrollo, definidas en el módulo “Gestionar Tareas (Mod-42)”, agrupadas por Sprint. Como se muestra en la Figura 47, por cada tarea definida, el módulo permite señalar el día del Sprint en que ésta se llevará a cabo. Adicionalmente, este módulo también genera un gráfico con el esfuerzo estimado del Sprint, que se muestra al final de la Figura 47. La línea roja del gráfico indica la planificación determinada para las tareas, y la línea azul señala la estimación ideal de las tareas. Finalmente, el Equipo de Desarrollo es el *Responsable* de este módulo, y el Arquitecto de la Información, el Scrum-UIA Master y el Dueño del Producto sólo tienen acceso a *Lectura*.



**Figura 47:** Módulo para estimar el esfuerzo diario de las tareas de desarrollo (Mod-51), junto con el gráfico del esfuerzo estimado de las tareas de desarrollo. La línea roja del gráfico indica la planificación de las tareas y la línea azul señala la estimación ideal de las tareas.

A continuación, se describen las entradas y la salida de este módulo:

### Entradas

- El proyecto seleccionado y activado en el módulo “Consultar Proyectos (Mod-11)”.
- Los Sprints definidos en el módulo “Agregar Sprint (Mod-32)”.
- Las tareas de desarrollo del Sprint seleccionado.
- La estimación del esfuerzo estimado de las tareas de desarrollo para cada día.

### Salida

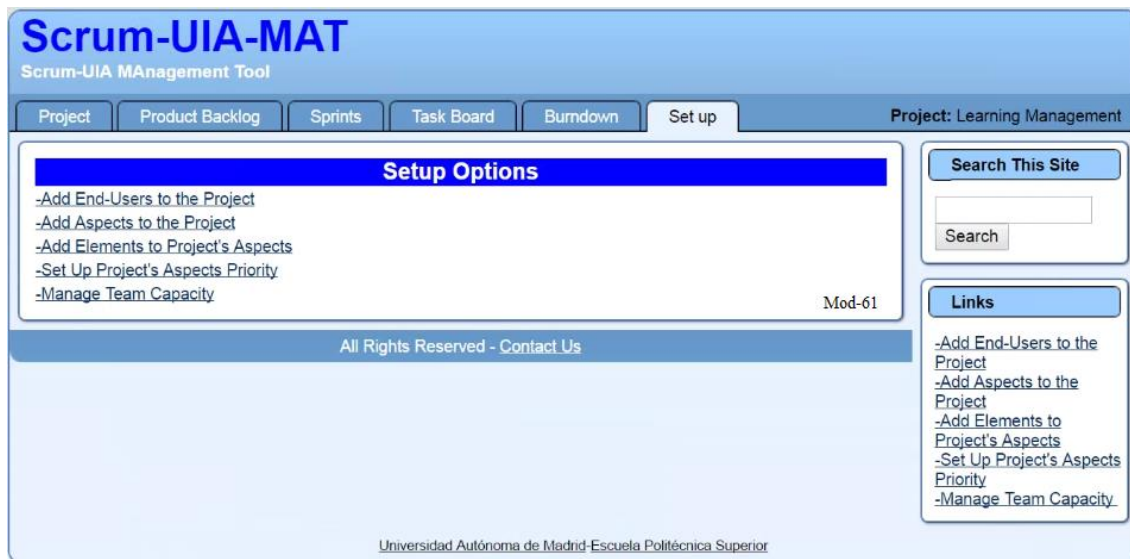
- La determinación del esfuerzo estimado de las tareas de desarrollo.

#### 3.5.3.6. Configuración (Set Up)

Esta opción agrupa los módulos que permiten realizar diversas configuraciones en los proyectos, tales como especificar los usuarios finales, detallar los aspectos y los elementos relevantes del proyecto, indicar las prioridades de los aspectos y determinar la capacidad del Equipo de Desarrollo. A continuación, se presentan en detalle los seis módulos de esta rama de navegación:

##### a) Opciones de Configuración (Mod-61)

En este módulo (ver Figura 48), se listan y organizan las diferentes opciones de configuración de la herramienta Scrum-UIA-MAT. El Arquitecto de la Información es el *Responsable* de este módulo y el Equipo de Desarrollo, el Scrum-UIA Master y el Dueño del Producto sólo tienen acceso a *Lectura*.



**Figura 48:** Módulo con las opciones de configuración (Mod-61).

Las salidas de este módulo corresponden a:

### Salidas

- El enlace para el módulo “Gestionar Usuarios Finales (Mod-62)”.
- El enlace para el módulo “Gestionar Aspectos (Mod-63)”.
- El enlace para el módulo “Gestionar Elementos (Mod-64)”.
- El enlace para el módulo “Configurar Prioridades de los Aspectos (Mod-65)”.
- El enlace para el módulo “Gestionar la Capacidad del Equipo (Mod-66)”.

### b) Gestionar Usuarios Finales (Mod-62)

Este módulo (ver Figura 49) permite consultar, editar e insertar los usuarios finales de cada proyecto (correspondiente a la Tarea 1.3.1 de Scrum-UIA). Como se muestra en la Figura 49, los usuarios finales se listan de acuerdo al proyecto seleccionado y activado en el módulo “Consultar Proyectos (Mod-11)”. Asimismo, en la parte izquierda de cada usuario final registrado en el módulo, se indica la opción para editar su nombre (opción *Edit*), y en la parte inferior del módulo se encuentra la funcionalidad para agregar un nuevo usuario final (opción *New*). El Arquitecto de la Información es el *Responsable* de este módulo, el Dueño del Producto tiene acceso a *Escritura*, y el Scrum-UIA Master y el Equipo de Desarrollo sólo tienen acceso a *Lectura*.

**Scrum-UIA-MAT**  
Scrum-UIA MAnagement Tool

Project | Product Backlog | Sprints | Task Board | Burndown | Set up | Project: Learning Management

### End-Users of the Project

	Id	Name
<a href="#">Edit</a>	5	Admin
<a href="#">Edit</a>	2	Admin & Instructor
<a href="#">Edit</a>	1	Instructor
<a href="#">Edit</a>	4	Observer
<a href="#">Edit</a>	3	Student
<a href="#">Edit</a>	10	Student & Instructor

Id

5

Name

Admin

idProject

1

New

Mod-62

All Rights Reserved - Contact Us  
Universidad Autónoma de Madrid-Escuela Politécnica Superior

Search This Site

Search

Links

- Add End-Users to the Project
- Add Aspects to the Project
- Add Elements to Project's Aspects
- Set Up Project's Aspects Priority
- Manage Team Capacity

**Figura 49:** Módulo para gestionar los usuarios finales (Mod-62).



Las entradas y la salida de este módulo corresponden a:

### Entradas

- El proyecto seleccionado y activado en el módulo “Consultar Proyectos (Mod-11)”.
- Los usuarios finales ingresados en este módulo.

### Salida

- La identificación de los usuarios finales de cada proyecto.

### c) Gestionar Aspectos (Mod-63)

En la Figura 50, se presenta el módulo para gestionar los aspectos relevantes del proyecto (correspondiente a la Tarea 1.4.1 de Scrum-UIA). Este módulo permite consultar, editar e insertar los aspectos relevantes de cada proyecto. Como se muestra en la Figura 50, los aspectos relevantes son listados de acuerdo al proyecto seleccionado y activado en el módulo “Consultar Proyectos (Mod-11)”. Además, el módulo permite editar y agregar los aspectos relevantes a través de las opciones *Edit* y *New*, respectivamente. Asimismo, el Arquitecto de la Información es el *Responsable* de este módulo, el Dueño del Producto tiene acceso a *Escritura*, y el Scrum-UIA Master y el Equipo de Desarrollo sólo tienen acceso a *Lectura*. Las entradas y la salida de este módulo corresponden a:

### Entradas

- El proyecto seleccionado y activado en el módulo “Consultar Proyectos (Mod-11)”.
- El nombre de los aspectos relevantes del proyecto.

### Salida

- La identificación de los aspectos relevantes del proyecto.

**Scrum-UIA-MAT**  
Scrum-UIA MAnagement Tool

Project | Product Backlog | Sprints | Task Board | Burndown | Set up | Project: Learning Management

### Relevant Aspects of the Project

	Id	Name
<a href="#">Edit</a>	3	Business Value
<a href="#">Edit</a>	2	Content
<a href="#">Edit</a>	1	Usability

id: 3  
name: Business Value  
[New](#)

Mod-63

All Rights Reserved - [Contact Us](#)

Universidad Autónoma de Madrid-Escuela Politécnica Superior

**Search This Site**

[Search](#)

**Links**

- Add End-Users to the Project
- Add Aspects to the Project
- Add Elements to Project's Aspects
- Set Up Project's Aspects Priority
- Manage Team Capacity

**Figura 50:** Módulo para gestionar los aspectos relevantes del proyecto (Mod-63).

#### d) Gestionar Elementos (Mod-64)

En la Figura 51, se presenta el módulo para consultar e insertar los elementos de los aspectos relevantes del proyecto (correspondiente a la Tarea 1.4.1 de Scrum-UIA). Como se muestra en la Figura 51, en este módulo se listan todos los proyectos que se encuentran definidos en la herramienta, donde los aspectos relevantes (ver campo *Aspects*) se listan de acuerdo al proyecto seleccionado (ver campo *Project*). Asimismo, los datos requeridos para insertar un nuevo elemento corresponden al nombre, la descripción y la prioridad del elemento (correspondientes a los campos *Name*, *Description* y *Priority*, respectivamente). Finalmente, el Arquitecto de la Información es el *Responsable* de este módulo, el Dueño del Producto tiene acceso a *Escritura*, y el Scrum-UIA Master y el Equipo de Desarrollo sólo tienen acceso a *Lectura*. Las entradas y la salida de este módulo corresponden a:

##### Entradas

- Los proyectos definidos en el módulo “Agregar Proyecto (Mod-12)”.
- Los aspectos relevantes del proyecto definidos en el módulo “Gestionar Aspecto (Mod-63)”.
- Los nombres de los elementos de los aspectos relevantes.
- Las descripciones de los elementos de los aspectos relevantes.
- Las prioridades de los elementos de los aspectos relevantes.

##### Salida

- La determinación de los elementos de cada aspecto relevante del proyecto.

The screenshot displays the 'Scrum-UIA-MAT' web application. At the top, there's a navigation bar with tabs: 'Project', 'Product Backlog', 'Sprints', 'Task Board', 'Burndown', and 'Set up'. The 'Project' tab is active, showing 'Project: Learning Management'. Below this, a form titled 'Add Elements of the Aspects' is visible. It includes a dropdown for 'Project' (set to 'Learning Management'), radio buttons for 'Aspects' (Business Value, Content, Usability), text fields for 'Name' and 'Description', a 'Priority' dropdown (set to 'High'), and an 'Add' button. At the bottom of the form are 'Accept' and 'Cancel' buttons. To the right of the form is a sidebar with a 'Search This Site' section and a 'Links' section containing several links like '-Add Aspects to the Project' and '-Add End-Users to the Project'. The footer of the application states 'All Rights Reserved - Contact Us' and 'Universidad Autónoma de Madrid-Escuela Politécnica Superior'.

**Figura 51:** Módulo para gestionar los elementos de los aspectos relevantes del proyecto (Mod-64).

**e) Configurar Prioridades de los Aspectos (Mod-65)**

Este módulo (ver Figura 52) permite establecer las prioridades de los aspectos relevantes del proyecto (correspondiente a la Tarea 1.4.1 de Scrum-UIA). Como se muestra en la Figura 52, este módulo lista todos los aspectos relevantes del proyecto, seleccionado y activado en el módulo “Consultar Proyectos (Mod-11)”, y permite priorizar a cada uno de ellos, utilizando la siguiente escala: Low, Medium y High. Finalmente, el Arquitecto de la Información es el *Responsable* de este módulo, el Dueño del Producto tiene acceso a *Escritura*, y el Scrum-UIA Master y el Equipo de Desarrollo sólo tienen acceso a *Lectura*.



**Figura 52:** Módulo para establecer las prioridades de los aspectos relevantes del proyecto (Mod-65).

Las entradas y la salida de este módulo corresponden a:

**Entradas**

- El proyecto seleccionado y activado en el módulo “Consultar Proyectos (Mod-11)”.
- Los aspectos relevantes del proyecto, definidos en el módulo “Gestionar Aspectos (Mod-63)”.
- Las prioridades de los aspectos relevantes.

**Salida**

- La determinación de las prioridades de cada aspecto relevante del proyecto.

**f) Gestionar la Capacidad del Equipo (Mod-66)**

Este módulo (ver Figura 53) permite determinar la disponibilidad de recursos del Equipo de Desarrollo (Tarea 3.1.2 de Scrum-UIA) y estimar su velocidad (Tarea 3.1.3 de Scrum-UIA). Como se puede observar en la Figura 53, en el apartado *Available Person-Days*, se listan los miembros del Equipo de Desarrollo con los datos referentes a los días-persona disponibles para el siguiente Sprint (ver *Next Sprint*) y los días-persona utilizados en el Sprint anterior (ver *Previous Sprint*). La información anterior permite conocer el total de días-persona disponibles del Equipo de Desarrollo para el siguiente y el anterior Sprint (ver *Total* en la Figura 53). De este modo, el Equipo de

Desarrollo puede determinar su velocidad estimada (en puntos de historias, ver campo *Estimated Velocity* en la Figura 53) para el siguiente Sprint, de acuerdo a su velocidad real (puntos de historia completados en el Sprint anterior, ver campo *Real Velocity* en la Figura 53) y su factor de dedicación (ver campo *Factor Dedication*). Finalmente, el Equipo de Desarrollo es el *Responsable* de este módulo, mientras que el Dueño del Producto y el Scrum-UIA Master sólo tienen acceso a *Lectura*.

Manage Team					
Available Person-Days					
	Id	Name	Next Sprint	Previous Sprint	Type
<a href="#">Edit</a>	2	Development Team	40	40	Development Team
<a href="#">Edit</a>	1	Information Architect	20	20	Information Architect
<a href="#">Edit</a>	6	NN	35	20	Development Team
<a href="#">Edit</a>	4	Product Owner	0	0	Product Owner
<a href="#">Edit</a>	3	Scrum Master	0	0	Scrum Master
Total			95	80	
Real Velocity:		<input type="text" value="70"/>			
Factor Dedication:		(Real Velocity / Available Previous Sprint)			<input type="text" value="0,9"/>
Estimated Velocity:		(Factor Dedication * Available Next Sprint)			<input type="text" value="85,5"/>

Add Team Member		
Name	<input type="text"/>	
Next Sprint	<input type="text"/>	
Previous Sprint	<input type="text"/>	
Type	<input type="text"/>	
<a href="#">Insert</a> <a href="#">Cancel</a>		

Mod-66

**Figura 53:** Módulo para gestionar la capacidad del Equipo de Desarrollo (Mod-66).

Las entradas y las salidas de este módulo corresponden a:

#### Entradas

- El proyecto seleccionado y activado en el módulo “Consultar Proyectos (Mod-11)”.
- Los miembros del Equipo de Desarrollo.
- Los días-persona disponibles de los miembros del Equipo de Desarrollo para el siguiente Sprint.

- Los día-persona utilizados por los miembros del Equipo de Desarrollo en el Sprint anterior.
- Velocidad real del Equipo de Desarrollo.

#### **Salidas**

- La identificación de la velocidad estimada del Equipo de Desarrollo.
- La identificación del factor de dedicación del Equipo de Desarrollo.
- El total de los días-persona disponibles y utilizados por el Equipo de Desarrollo en los Sprints.

### **3.6. Recapitulación**

A lo largo de este capítulo, se ha presentado una metodología para integrar la AI en un proceso ágil de desarrollo centrado en el usuario, llamada Scrum-UIA, que permite la implementación de los requisitos a través de un modelo de desarrollo incremental e iterativo. Dicha metodología prescribe 16 actividades, que se organizan en cuatro grupos principales: Investigación Contextual, Gestión de Requisitos, Planificación del Sprint, e Inspección y Mejora Continua. Además, para cada actividad se especifican los objetivos, los productos y las tareas requeridas para llevarlas a cabo. Se prescribieron un total de 29 tareas y 31 productos.

Asimismo, se ha ideado un nuevo método de priorización de requisitos, llamado QMPSR, adaptado a la metodología Scrum-UIA, que permite dirigir el proceso de priorización (realizado en el Grupo de Actividades "Gestión de Requisitos") a través de las prioridades de la usabilidad, de la AI y del valor de negocio. Estas prioridades son capturadas de manera formal (durante el Grupo de Actividades "Investigación Contextual"), mediante el uso de elementos y aspectos cualitativos que definen el nivel de relevancia de los requisitos. Esto abre la oportunidad para analizar, discutir y verificar colectivamente la pertinencia de los requisitos demandados en cada Sprint del proyecto, y gestionar de manera dinámica el proceso de priorización.

Además, se ha propuesto una técnica, llamada InterArch-T, que permite promover el desarrollo incremental de los requisitos en Scrum-UIA a través de las prioridades de la AI. Dicha técnica se ha implementado mediante la elaboración de una herramienta CASE, llamada InterArch, que permite diseñar representaciones conceptuales de alto nivel del contenido, elaboradas por el Arquitecto de la Información, que evolucionan hacia el dominio de la solución (en diagramas de clases UML), y facilitan el desarrollo incremental en Scrum-UIA, dirigido por la AI. En ese sentido, se permite dar continuidad al resto de las tareas de desarrollo específicas, que el Equipo de Desarrollo debe realizar durante la Ejecución del Sprint, y aumentar la automatización e interoperabilidad en la implementación de los requisitos.

Por otro lado, se ha diseñado una herramienta Web, llamada Scrum-UIA-MAT, que proporciona un sistema de gestión de proyectos basado en Scrum-UIA. Esta herramienta provee diversas funcionalidades para promover el desarrollo ágil de la AI, gestionar las prioridades de los usuarios finales del proyecto y conducir el proceso de priorización de los requisitos mediante el uso de elementos cualitativos, relacionados con las preferencias de los proyectos.

Finalmente, se puede afirmar que la solución propuesta en este capítulo permite una integración eficiente de la AI dentro de un proceso ágil de desarrollo centrado en el usuario, considerando diversos elementos necesarios para integrar el DCU en los entornos cambiantes. No obstante, se hace también necesario identificar la idoneidad de las características de Scrum-UIA, así como las de su herramienta Scrum-UIA-MAT, para desarrollar aplicaciones interactivas en entornos ágiles, teniendo en cuenta las prioridades de los usuarios finales durante las diferentes fases. Por lo tanto, en el siguiente capítulo se realizará una verificación de Scrum-UIA, junto con el soporte de Scrum-UIA-MAT, para comprobar la capacidad de sus actividades y tareas, lo que permitirá corroborar la hipótesis de partida **H2.1**: Es posible idear una metodología para integrar la Arquitectura de la Información en un proceso ágil de desarrollo centrado en el usuario.

# Capítulo 4: Verificación

---

**E**n este capítulo, se describe la verificación de la idoneidad de las actividades y tareas de la metodología Scrum-UIA, y de su herramienta Scrum-UIA-MAT, mediante la aplicación de un ejemplo práctico de desarrollo. En primer lugar, en la sección 4.1, se introducen los aspectos generales del método de verificación. Posteriormente, en la sección 4.2, se presenta la descripción de la aplicación seleccionada para realizar el proceso de verificación. Seguidamente, en la sección 4.3, se describe la planificación de la ejecución de las actividades. Finalmente, en la sección 4.4, se detallan la realización de las actividades y de las tareas propuestas en la metodología Scrum-UIA, junto con el soporte de Scrum-UIA-MAT.

## 4.1. Método de Verificación

La verificación de la metodología Scrum-UIA se plantea mediante la aplicación de un ejemplo práctico de desarrollo, centrándose en la implementación de las actividades de Scrum-UIA (descritas en la sección 3.2), a través del apoyo de la herramienta Scrum-UIA-MAT (presentada en la sección 0), para el despliegue de un ejemplo concreto de aplicación. De este modo, las tareas soportadas por la herramienta Scrum-UIA-MAT se realizarán mediante los módulos correspondientes y se elaborarán los productos involucrados en la ejecución de dichas tareas, lo que entregará una idea de la idoneidad de las actividades de Scrum-UIA.

## 4.2. Aplicación de Gestión de Aprendizaje para el Caso de Estudio

La aplicación Web escogida para efectuar el proceso de verificación es un SI de gestión de aprendizaje para una universidad. El objetivo de esta aplicación es que los procesos principales para la inscripción, la matrícula y el seguimiento de los cursos de una universidad se realicen mediante un entorno Web de manera electrónica. Asimismo, los usuarios finales de este SI corresponden a estudiantes, instructores (profesores), observadores y personal administrativo de la universidad. Por lo tanto, se requiere la implementación de una aplicación Web con las siguientes prestaciones:

- Que los instructores y administradores puedan crear, editar y consultar los cursos de la universidad.
- Que los instructores y estudiantes puedan gestionar los cursos en un entorno integrado.
- Que los estudiantes puedan crear, editar y consultar las actividades de evaluación de cada curso.

- Que los administradores puedan configurar los perfiles de los estudiantes y de los profesores.
- Que los administradores puedan gestionar las inscripciones en los cursos.

Por otro lado, una de las principales prioridades de este proyecto es que se consideren, desde etapas iniciales, las características de los usuarios finales, con el objetivo de que estos elementos conduzcan el desarrollo de las diferentes funcionalidades del SI de gestión de aprendizaje. Por lo tanto, se ha conformado un equipo de trabajo informático real para implementar la metodología Scrum-UIA, junto con el soporte de la herramienta Scrum-UIA-MAT, con el fin de llevar a cabo el proyecto. En la Tabla 25, se presentan los cinco miembros del equipo de trabajo y los diferentes roles utilizados dentro de la metodología Scrum-UIA.

Roles de Scrum-UIA	Miembros del Equipo Informático	Abreviaturas
Arquitecto de la Información	Luis Rojas	L.R.
Dueño del Producto	Juan Pérez	J.P.
Equipo de Desarrollo	Mario Reyes	M.R.
Equipo de Desarrollo	Adams Galkus	A.G.
Scrum-UIA Master	María Gaete	M.G.

**Tabla 25:** Roles de Scrum-UIA, utilizados por los miembros del equipo de trabajo informático.

### 4.3. Planificación de Actividades

Teniendo en cuenta el caso de estudio descrito anteriormente, se ha planificado la ejecución de las actividades de Scrum-UIA. De esta manera, las actividades y tareas de Scrum-UIA (prescritas en la sección 3.2.2) se ejecutan de forma secuencial, de acuerdo a las interdependencias entre ellas, con la finalidad de alcanzar el objetivo planteado. Además, en la ejecución de las tareas, se consideran la generación de los productos definidos y el apoyo de los módulos involucrados de la herramienta Scrum-UIA-MAT. De este modo, la primera tarea que se lleva a cabo es la Tarea 1.1.1, y la última es la Tarea 4.3.1.

La Figura 54 presenta los productos que se generan en las actividades, e identifica los módulos de la herramienta Scrum-UIA-MAT relacionados con estos productos (en la parte superior azul del producto). Por un lado, en el Grupo de Actividades de la Investigación Contextual, se genera un total de nueve productos (P1, P2, P3, P4, P5, P6, P7, P8 y P9), con el fin de obtener información de los usuarios finales, del contenido y del contexto del proyecto. Destaca el producto P4 de la Actividad 1.2 (Identificación y análisis de contenido) que permite describir, de manera gráfica, las prioridades y las relaciones de los contenidos más importantes del proyecto. También destacan los productos P8 y P9, que son los principales resultados de la Actividad 1.4 (Generación del panorama general del proyecto). Por un lado, en el producto P8, se identifican los elementos y los aspectos



relevantes del proyecto, generados por los módulos Mod-63 y Mod-64 (para los aspectos y los elementos, respectivamente), y que posteriormente son utilizados como fuente principal de información en el módulo Mod-22. Por otro lado, en el producto P9, se determinan los criterios de aceptación de los usuarios finales (estudiantes, instructores, administradores y observadores), que también son utilizados como fuente principal de información en el módulo Mod-22.

Asimismo, en el Grupo de Actividades de la Gestión de Requisitos, se genera un total de ocho productos (P10, P11, P12, P13, P14, P15, P16 y P17) que tienen como objetivo generar una Lista del Producto con sus elementos descritos, estimados y priorizados, siendo éstos la base para iniciar la Planificación del Sprint. Entre estos productos destacan los P12, P13, P14 y P15 de la Actividad 2.2 (Generación final de requisitos) que identifican los requisitos del proyecto. Por ejemplo, en el producto P12, se describe el catálogo final revisado de los requisitos del proyecto, que sirve como fuente de información para los módulos Mod-21 y Mod-22. De manera similar, el producto P13 identifica los usuarios finales, la capacidad y el valor de negocio de los requisitos de la Lista del Producto, generados por el Mod-22, y se utiliza como fuente de información en el Mod-21. Finalmente, mientras que en el producto P14 se determina la estimación de los requisitos en puntos de historia y días-persona, en el producto P15 se identifican las priorizaciones entregadas por el Dueño del Producto y el Arquitecto de la Información para los requisitos del proyecto. Ambos productos se generan en el Mod-22, y son fuente de información para el Mod-21.

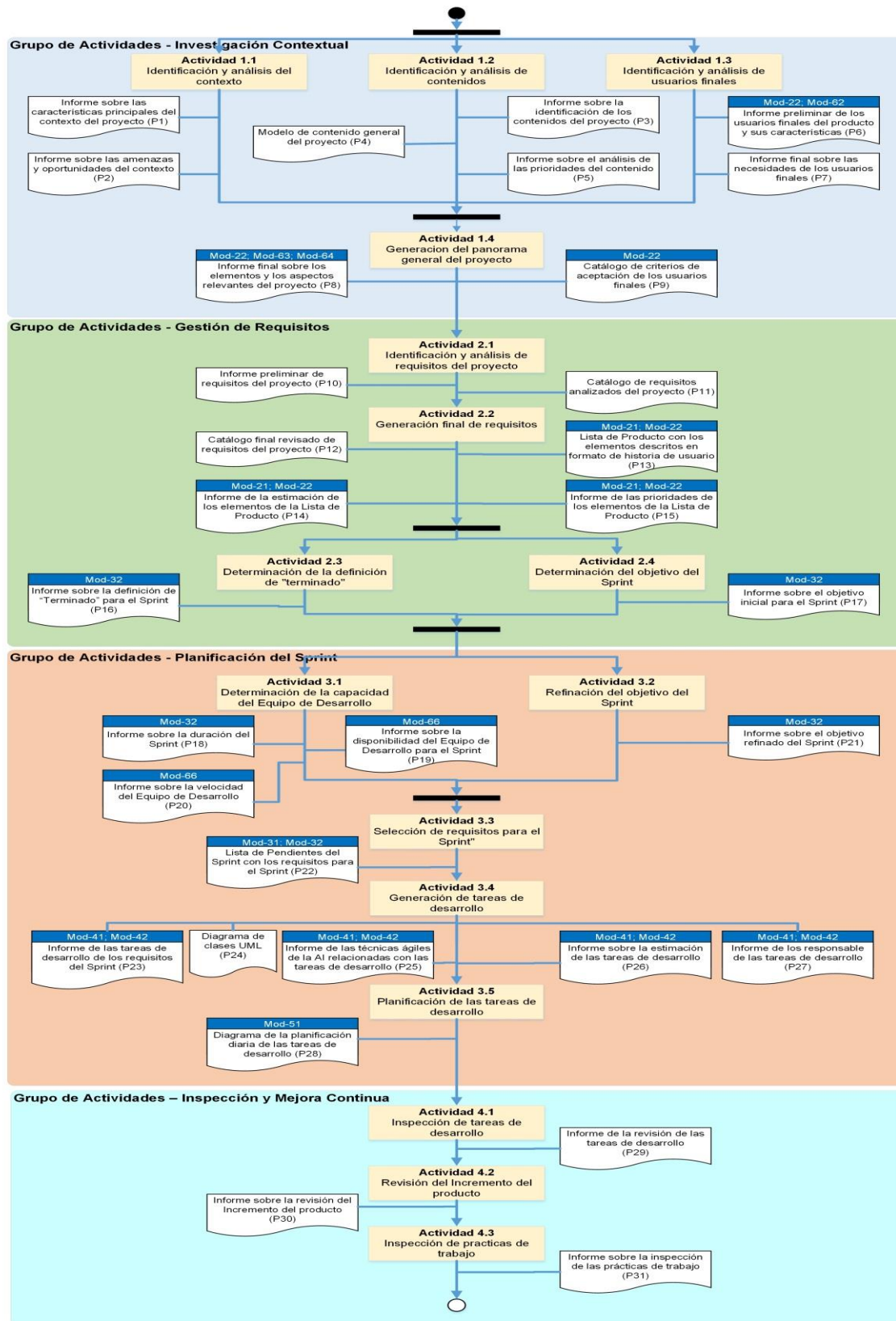
Respecto al Grupo de Actividades de la Planificación del Sprint, éste tiene un total de once productos (P18, P19, P20, P21, P22, P23, P24, P25, P26, P27 y P28) que tienen como objetivo describir la Lista de Pendientes del Sprint (es decir, las tareas de desarrollo para implementar los requisitos seleccionados para el Sprint) y planificar el esfuerzo de las tareas durante los días del Sprint. Destacan los productos P18 y P20 de la Actividad 3.1 (Determinación de la capacidad del Equipo de Desarrollo) que identifican la duración del Sprint y la velocidad del Equipo de Desarrollo para el siguiente Sprint, respectivamente. Estos productos se generan en los módulos Mod-32 y Mod-66 (P18 y P20, respectivamente). Asimismo, los productos P23, P25 y P26 de la Actividad 3.4 (Generación de tareas de desarrollo) identifican las tareas de desarrollo, las técnicas ágiles de la AI y la estimación de las tareas, respectivamente. Todos estos productos se crean en el módulo Mod-42, y son fuente de información del módulo Mod-41.

Por último, el Grupo de Actividades de Inspección y Mejora Continua tiene un total de tres productos (P29, P30 y P31) que permiten fomentar el DCU y la participación de los usuarios finales en las instancias de inspección y de evaluación de Scrum-UIA.

Como se puede observar en la Figura 54, algunas tareas y actividades de Scrum-UIA no cuentan con el soporte de Scrum-UIA-MAT. Esto se debe a que las funcionalidades de

Scrum-UIA-MAT se centran, principalmente, en las tareas y actividades que producen datos que se requieren sistematizar, para proveer continuidad en el proceso de Scrum-UIA.

Por ejemplo, la “Actividad 1.1. Identificación y análisis del contexto” no cuenta con soporte de Scrum-UIA-MAT, debido a que se generan resultados intermedios (datos sobre la identificación del contexto del proyecto) que no se necesitan sistematizar para continuar con las otras actividades de Scrum-UIA. Por el contrario, la “Actividad 1.4 Generación del panorama general del proyecto” cuenta con el soporte de Scrum-UIA-MAT, debido a que se generan resultados finales (los elementos y aspectos relevantes del proyecto) que posteriormente son requeridos en tareas y actividades para continuar correctamente el proceso de Scrum-UIA (por ejemplo, en la “Actividad 2.2 Generación final de requisitos”).



**Figura 54:** Productos creados de acuerdo a las actividades que los generan y la identificación de los módulos que se relacionan.

## 4.4. Ejecución de las Actividades de Verificación

Una vez planificadas las actividades, en esta sección se describen en detalle la ejecución de las actividades y de las tareas propuestas en la metodología Scrum-UIA, y el soporte de la herramienta Scrum-UIA-MAT.

### 4.4.1. Actividad 1.1. Identificación y análisis del contexto

Mediante esta actividad, se realizará un estudio para identificar y analizar el contexto de la universidad. Para ello, se obtendrá información de la cultura corporativa, los objetivos (a corto y largo plazo), los competidores potenciales y directos, el ámbito económico y tecnológico, a través de entrevistas no estructuradas y reuniones de investigación con los stakeholders de la universidad (trabajadores, directivos, accionistas, proveedores, sindicato, estudiantes, gobierno corporativo). Además, se revisará la información recabada, con el fin de analizar las amenazas y las oportunidades del contexto de la universidad.

Al realizar esta actividad, se identifican las características principales del contexto de la universidad, y se describen sus amenazas y oportunidades.

#### 4.4.1.1. Tarea 1.1.1. Identificar contexto del proyecto

**Descripción:** Se identificarán las particularidades más esenciales del contexto de la universidad.

Inicialmente, se realizarán entrevistas no estructuradas y reuniones con los stakeholders de la universidad, con el fin de identificar las fuentes principales de información.

**Información de entrada:** La principal información de entrada corresponde a los objetivos, los presupuestos, los programas, la infraestructura tecnológica, los recursos humanos y la cultura corporativa de la universidad (E1).

Esta información se obtiene principalmente mediante investigaciones de antecedentes, presentaciones, reuniones de investigación y entrevistas no estructuradas con los stakeholders de la universidad.

**Producto de salida:** El resultado de la Tarea 1.1.1 corresponde a la identificación de las características principales del contexto de la universidad, en base a las entrevistas no estructuradas y las reuniones de investigación con los stakeholders. A continuación, se presenta en detalle este producto:

### **P1. Informe sobre las características principales del contexto**

**Autor:** Arquitecto de la Información (L.R.)

**Fecha:** 27 Julio 2016

**CC1:** Se ha incrementado la participación de la universidad en los estudiantes de entrada.

**CC2:** La competencia de otras ofertas educativas superiores es alta.

**CC3:** Se ha diversificado el número de titulaciones ofertadas.

**CC4:** Se ha incrementado el número de estudiantes de postgrado.

**CC5:** Los estudiantes y trabajadores provienen de diferentes culturas.

**Producto de Verificación 1:** Informe sobre las características principales del contexto.

#### **4.4.1.2. Tarea 1.1.2. Analizar exigencias y oportunidades del contexto**

**Descripción:** Una vez identificadas las características más importantes del contexto de la universidad, se realizarán reuniones y entrevistas no estructuradas con los stakeholders, con el fin de determinar las amenazas y oportunidades del contexto de la universidad.

En general, estas técnicas se llevan a cabo para analizar en conjunto cuáles son las principales exigencias y oportunidades del contexto de la universidad.

**Información de entrada:** La principal fuente de información de esta tarea es el producto “Informe sobre las características principales del contexto” (P1), generado en la Tarea 1.1.1.

**Productos de salida:** El resultado de esta tarea corresponde a la determinación de las principales amenazas y oportunidades del contexto de la universidad. A continuación, se presenta en detalle este producto:

### **P2. Informe sobre las amenazas y oportunidades del contexto**

**Autor:** Arquitecto de la Información (L.R.)

**Fecha:** 27 Julio 2016

#### **Amenazas del contexto**

**A1:** La competencia de otras universidades e instituciones próximas.

**A2:** La disminución de la población en edad académica y considerable emigración de estudiantes a otras universidades.

**A3:** La falta de presupuesto y financiamiento de proyectos específicos.

## **P2. Informe sobre las amenazas y oportunidades del contexto**

**Autor:** Arquitecto de la Información (L.R.)

**Fecha:** 27 Julio 2016

### **Oportunidades del contexto**

- O1:** El uso de las nuevas tecnologías ofrece unas posibilidades de proyección y gestión del aprendizaje aún no suficientemente explotadas.
- O2:** Entorno dinámico y complejo que obliga al cambio continuo.
- O3:** Impacto de las tecnologías en las prácticas educativas.
- O4:** Desarrollo de las nuevas tecnologías.
- O5:** Elevada tasa de inmigración.
- O6:** Mayor necesidad de formación continuada de los egresados.
- O7:** Existencia de recursos internacionales para la investigación.

**Producto de Verificación 2:** Informe sobre las amenazas y oportunidades del contexto.

### **4.4.2. Actividad 1.2. Identificación y análisis de contenidos**

En esta actividad se realizará un estudio de los contenidos relacionados con el proyecto. Para ello, se revisarán y analizarán las imágenes, los archivos de video, los documentos, los gráficos, los archivos de audio y cualquier información asociada a la gestión de conocimiento de la universidad.

Al finalizar esta actividad, se identifican los contenidos más importantes que se deben considerar para gestionar el conocimiento de la universidad, y se diseña un modelo de contenido general del SI de gestión de aprendizaje, que identifica las prioridades y componentes del contenido.

#### **Tarea 1.2.1. Identificar contenidos del proyecto**

**Descripción:** Se identifican los documentos y los objetos de contenido, relacionados con la gestión de conocimiento de la universidad.

El Arquitecto de la Información lleva a cabo una investigación de antecedentes para recopilar los objetos y los elementos de contenido, relacionados con la gestión de aprendizaje de la universidad. Asimismo, el Arquitecto de la Información utiliza la herramienta InterArch para generar un modelo general de los elementos de contenido, relacionado con la gestión de conocimiento de la universidad. El objetivo de este modelo de contenido es obtener un panorama general del proyecto, relacionado con los contenidos de la universidad.

**Información de entrada:** La principal información de entrada son los documentos y los objetos de contenido (en diferentes formatos), relacionados con la gestión de conocimiento de la universidad (E2).

En general, esta información se obtiene a través de entrevistas no estructuradas y reuniones con los stakeholders de la universidad.

**Productos de salida:** Esta tarea genera un informe que identifica los contenidos actuales, relacionados con la gestión de aprendizaje (P3) y un modelo de contenido general (P4). A continuación, se describen en detalle estos productos:

<b>P3. Informe sobre la identificación de los contenidos del proyecto</b>	
<b>Autor:</b> Arquitecto de la Información (L.R.)	
<b>Fecha:</b> 27 Julio 2016	
<b>CIP1:</b> Datos personales de los estudiantes.	
<b>Descripción:</b> Descripción detallada de los datos personales de los estudiantes de la universidad.	
<b>Ubicación:</b> C:\FileStudent\Dapees.xlsx	
<b>CIP2:</b> Catálogo de los cursos ofrecidos.	
<b>Descripción:</b> Identificación y descripción de los cursos ofrecidos por la universidad.	
<b>Ubicación:</b> C:\FileStudent\Cacuof.xlsx	
<b>CIP3:</b> Fotografías de los estudiantes e instructores.	
<b>Descripción:</b> Carpeta con fotografías en forma JPG de los estudiantes e instructores de la universidad.	
<b>Ubicación:</b> C:\File\picturesStudentsinstructos	
<b>CIP4:</b> Actividades asociadas a los cursos.	
<b>Descripción:</b> Registro de las actividades realizadas en los cursos de la universidad.	
<b>Ubicación:</b> C:\File\cor\accu.xls	
<b>CIP5:</b> Datos personales de los instructores.	
<b>Descripción:</b> Descripción detallada de los datos personales de los instructores de la universidad.	
<b>Ubicación:</b> C:\FileStudent\Dapein.xlsxci	
<b>CIP6:</b> Resultados de las actividades de evaluación.	
<b>Descripción:</b> Registro de las calificaciones y observaciones de las actividades de evaluación de los estudiantes.	

**Ubicación:** C:\FileStudent\evaacev.xlsx

**CIP7:** Matricula de los estudiantes.

**Descripción:** Registro de los datos de matrícula por cada estudiante.

**Ubicación:** C:\FileStudent\maes.xlsx

**Producto de Verificación 3:** Informe sobre la identificación de los contenidos del proyecto.

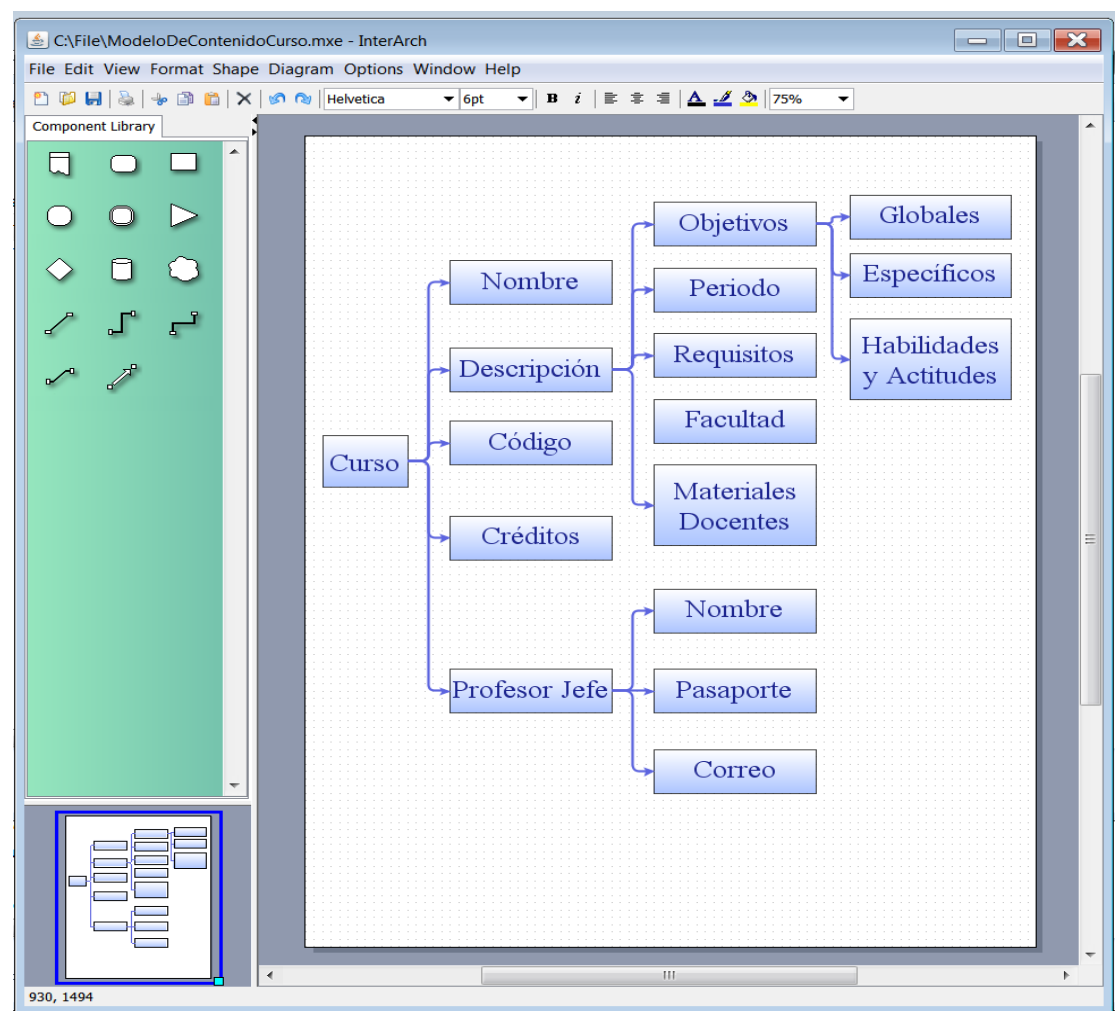
#### P4. Modelo de contenido general del proyecto

**Autor:** Arquitecto de la Información (L.R.)

**Fecha:** 27 Julio 2016

**MC1:** Modelo de contenido de los elementos relacionados con los cursos.

**Descripción:** Se describen los elementos de contenido de los que se componen los cursos ofrecidos por la universidad.



**Producto de Verificación 4:** Informe sobre la identificación de los contenidos del proyecto.



#### 4.4.2.1. Tarea 1.2.2. Analizar prioridades del contenido

**Descripción:** Una vez identificados los contenidos actuales que tiene la universidad para la gestión de aprendizaje, se realizarán reuniones y entrevistas no estructuradas con los stakeholders, con el fin de identificar y determinar los contenidos más relevantes, que deberían manipular la nueva aplicación para la gestión de aprendizaje.

**Información de entrada:** La principal fuente de esta tarea es el producto “Informe sobre la identificación de los contenidos del proyecto” (P3) y el “Modelo de contenido general del proyecto” (P4), ambos generados en la Tarea 1.2.1.

**Productos de salida:** El resultado de esta tarea corresponde a la determinación de los contenidos (y su prioridad) que debería manipular la aplicación para la gestión de aprendizaje de la universidad. A continuación, se presenta en detalle este producto:

<b>P5. Informe sobre el análisis de las prioridades del contenido</b>	
<b>Autor:</b> Arquitecto de la Información (L.R.)	
<b>Fecha:</b> 27 Julio 2016	
<b>CP1:</b> Contenido de las entidades de conocimiento.	
<b>Descripción:</b> Información relacionada con el contenido de las entidades de conocimiento (cursos, módulos, actividades).	
<b>Prioridad:</b> 25.	
<b>CP2:</b> Descripción de las entidades de conocimiento.	
<b>Descripción:</b> Datos relacionados con la descripción de las entidades de conocimiento.	
<b>Prioridad:</b> 20.	
<b>CP3:</b> Calificaciones de las actividades de evaluación.	
<b>Descripción:</b> Valoraciones obtenidas por los estudiantes en las actividades de evaluación.	
<b>Prioridad:</b> 15.	
<b>CP4:</b> Datos personales de los estudiantes.	
<b>Descripción:</b> Detalles de los datos personales de los estudiantes.	
<b>Prioridad:</b> 12.	
<b>CP5:</b> Datos generales de los instructores.	
<b>Descripción:</b> Detalles de los datos personales de los instructores.	
<b>Prioridad:</b> 11.	
<b>CP6:</b> Retroalimentación de las evaluaciones de los estudiantes.	

<b>P5. Informe sobre el análisis de las prioridades del contenido</b>	
<b>Autor:</b> Arquitecto de la Información (L.R.)	
<b>Fecha:</b> 27 Julio 2016	
<b>Descripción:</b> Comentarios y observaciones de los instructores, respecto a las actividades de evaluación realizadas por los estudiantes.	
<b>Prioridad:</b> 22.	
<b>CP7:</b> Trabajos realizados por los estudiantes.	
<b>Descripción:</b> Registro de los trabajos, llevados a cabo por los estudiantes en las diferentes entidades de conocimiento.	
<b>Prioridad:</b> 16.	
<b>CP8:</b> Registro de las actividades desarrolladas por los estudiantes.	
<b>Descripción:</b> Registro de las interacciones de los estudiantes, durante las entidades de conocimiento.	
<b>Prioridad:</b> 18.	
<b>CP9:</b> Registro de las comunicaciones entre estudiantes y profesores.	
<b>Descripción:</b> Detalles de las conversaciones, realizadas por los estudiantes y los profesores.	
<b>Prioridad:</b> 6.	
<b>CP10:</b> Preferencias de los estudiantes.	
<b>Descripción:</b> Registro de las predilecciones y cualidades relevantes de los estudiantes.	
<b>Prioridad:</b> 30.	

**Producto de Verificación 5:** Informe sobre el análisis de las prioridades del contenido.

#### 4.4.3. Actividad 1.3. Identificación y análisis de usuarios finales

Mediante esta actividad se efectuarán estudios de los usuarios finales del SI para la gestión de aprendizaje de la universidad. Para ello, se recabará información de los stakeholders de la universidad, quienes podrán indicar las características y las necesidades principales de los usuarios finales.

Al finalizar esta actividad, se identifican los usuarios finales del SI para la gestión de aprendizaje de la universidad. Además, se determinan cuáles son las necesidades principales de los usuarios finales.

#### 4.4.3.1. Tarea 1.3.1. Identificar usuarios finales del producto

**Descripción:** Se identificarán los usuarios finales del SI para la gestión de aprendizaje de la universidad.

Como punto de partida, se realizarán reuniones y entrevistas no estructuradas con los stakeholders para determinar los usuarios finales del SI. Asimismo, en la realización de esta tarea, se utilizará el módulo “Gestionar Usuarios Finales (Mod-62)” de Scrum-UIA-MAT (sección 3.5.3.6). Este módulo permite insertar, editar y consultar los usuarios finales. La Figura 55 presenta los usuarios finales del SI de aprendizaje, gestionados a través del módulo “Gestionar Usuarios Finales (Mod-62)”. Como se puede observar en la Figura 55, los usuarios finales del SI de gestión de aprendizaje de la universidad corresponden a Estudiante, Instructor, Administrador y Observador. Además, para cada uno de ellos, se identifican sus principales características.

**Scrum-UIA-MAT**  
Scrum-UIA MAnagement Tool

Project | Product Backlog | Sprints | Task Board | Burndown | Set up | Project: Gestión de Aprendizaje

End-Users of the Project			
	Id	Name	Description
<a href="#">Edit</a>	33	Administrador	Rango de edad entre 20-60 años. Demanda información la matricula de los estudiantes y gestion de las asignaturas. Conocimiento en computación medio-baja
<a href="#">Edit</a>	31	Estudiante	Rango de edad entre 17-55 años. Demanda información sobre la entidades de conocimiento y retroalimentación de las evaluaciones. Conocimiento en computación intermedio-alto.
<a href="#">Edit</a>	32	Instructor	Rango de edad entre 22-65 años. Demanda información sobre la gestión de las asignaturas. Conocimiento en computación intermedio.
<a href="#">Edit</a>	34	Observador	Rango de edad entre 35-70 años. Demanda información sobre progreso de los estudiantes. Conocimiento en computación medio-bajo.

Name   
Description   
[Insert](#) [Cancel](#) Mod-62

Search This Site  
  
[Search](#)

**Links**

- Add End-Users to the Project
- Add Aspects to the Project
- Add Elements to Project's Aspects
- Set Up Project's Aspects Priority
- Manage Team Capacity

All Rights Reserved - [Contact Us](#)  
Universidad Autónoma de Madrid-Escuela Politécnica Superior

**Figura 55:** Gestión de los usuarios finales mediante el uso del módulo “Gestionar Usuarios Finales (Mod-62)” de Scrum-UIA-MAT.

**Información de entrada:** La principal fuente de entrada son los usuarios finales y sus características (E3).

**Producto de salida:** El producto de salida corresponde a un Informe preliminar de los usuarios finales del proyecto y sus características (P6), que se puede consultar a través del módulo “Gestionar Usuarios Finales (Mod-62)”. A continuación, se presentan los detalles de este producto de salida:

<p><b>P6. Informe preliminar de los usuarios finales del proyecto y sus características</b></p> <p><b>Autor:</b> Arquitecto de la Información (L.R.)</p> <p><b>Fecha:</b> 27 Julio 2016</p>
<p><b>UF31:</b> Estudiante.</p> <p><b>Descripción:</b> Rango de edad entre 17-55 años. Demanda información sobre las entidades de conocimiento y retroalimentación de las evaluaciones. Conocimiento en informática intermedio-alto.</p>
<p><b>UF32:</b> Instructor.</p> <p><b>Descripción:</b> Rango de edad entre 22-65 años. Demanda información sobre la gestión de las asignaturas. Conocimiento en informática intermedio.</p>
<p><b>UF33:</b> Administrador.</p> <p><b>Descripción:</b> Rango de edad entre 20-60 años. Demanda información sobre la matrícula de los estudiantes y gestión de las asignaturas. Conocimiento en informática intermedio-bajo.</p>
<p><b>UF34:</b> Observador.</p> <p><b>Descripción:</b> Rango de edad entre 35-70 años. Demanda información sobre el progreso de los estudiantes. Conocimiento en informática intermedio-bajo.</p>

**Producto de Verificación 6:** Informe preliminar de los usuarios finales del proyecto y sus características.

#### 4.4.3.2. Tarea 1.3.2. Analizar necesidades de los usuarios finales

**Descripción:** Una vez identificados los usuarios finales en la tarea anterior, se realizarán entrevistas no estructuradas y reuniones con ellos, para determinar sus necesidades e intereses respecto al SI para la gestión de aprendizaje de la universidad.

**Información de entrada:** La principal fuente de esta tarea es el Informe preliminar de los usuarios finales del proyecto y sus características (P6), generado en la Tarea 3.1.1.

**Producto de salida:** El resultado de esta tarea corresponde a la determinación de las necesidades de los usuarios finales (P7), que se deben considerar en la aplicación para la gestión de aprendizaje de la universidad. A continuación, se presenta en detalle este producto:

### **P7. Informe final sobre las necesidades de los usuarios finales**

**Autor:** Arquitecto de la Información (L.R.)

**Fecha:** 28 Julio 2016

**NUF31.1:** Estudiante.

**Necesidad:** Revisar la descripción y el contenido de las entidades de conocimientos (cursos, grupos, módulos).

**Descripción:** Estudiantes requieren consultar el detalle y los requisitos de cada entidad de conocimiento.

**NUF31.2:** Estudiante.

**Necesidad:** Consultar las actividades de evaluación.

**Descripción:** Estudiantes demandan revisar el proceso de evaluación y obtener detalles de la retroalimentación de las actividades de evaluación.

**NUF31.3:** Estudiante.

**Necesidad:** Gestionar las fuentes de información.

**Descripción:** Estudiantes requieren manipular los archivos utilizados y generados en cada entidad de conocimiento mediante un repositorio centralizado.

**NUF32.2:** Instructor.

**Necesidad:** Gestionar las entidades de conocimientos (cursos, grupos, módulos).

**Descripción:** Instructores demandan gestionar (crear, editar, consultar y organizar) las entidades de conocimiento.

**NUF32.3:** Instructor.

**Necesidad:** Gestionar las evaluaciones.

**Descripción:** Instructores requieren gestionar (crear, editar y consultar) las actividades de evaluación y manipular los resultados.

**NUF32.4:** Instructor.

**Necesidad:** Gestionar las instancias de discusión.

**Descripción:** Instructores solicitan generar y coordinar instancias de discusión, relacionadas con las entidades de conocimiento.

**NUF32.5:** Instructor.

**Necesidad:** Revisar las interacciones y el progreso de los estudiantes.

**Descripción:** Instructores requieren consultar las interacciones de los estudiantes en

<b>P7. Informe final sobre las necesidades de los usuarios finales</b>	
<b>Autor:</b> Arquitecto de la Información (L.R.)	
<b>Fecha:</b> 28 Julio 2016	
las entidades de conocimiento.	
<b>NUF33.1:</b> Administrador.	
<b>Necesidad:</b> Gestionar las matrículas de los estudiantes.	
<b>Descripción:</b> Administradores demandan configurar la matrícula de los estudiantes.	
<b>NUF33.2:</b> Administrador.	
<b>Necesidad:</b> Gestionar los perfiles de los estudiantes y los instructores.	
<b>Descripción:</b> Administradores requieren configurar los datos de los perfiles de los estudiantes e instructores.	
<b>NUF33.3:</b> Administrador.	
<b>Necesidad:</b> Consultar la composición y los detalles de las entidades de conocimiento.	
<b>Descripción:</b> Administradores solicitan revisar los datos generales de las entidades de conocimiento.	
<b>NUF34.1:</b> Observador.	
<b>Necesidad:</b> Consultar la descripción y el contenido de las entidades de conocimiento.	
<b>Descripción:</b> Observadores requieren revisar los datos generales de las entidades de conocimiento de los estudiantes que supervisan.	
<b>NUF34.2:</b> Observador.	
<b>Necesidad:</b> Consultar los anuncios y las comunicaciones de las entidades de conocimiento.	
<b>Descripción:</b> Observadores exigen revisar los anuncios y comunicaciones de las entidades de conocimiento de los estudiantes supervisados.	
<b>NUF34.3:</b> Observador.	
<b>Necesidad:</b> Comunicarse con los instructores.	
<b>Descripción:</b> Observadores requieren enviar mensajes a los instructores de los estudiantes que supervisan.	

**Producto de Verificación 7:** Informe final sobre las necesidades de los usuarios finales.

#### 4.4.4. Actividad 1.4. Generación del panorama general del proyecto

Una vez identificados y analizados el contexto, el contenido y los usuarios finales del SI para la gestión de aprendizaje de la universidad, se procederá a analizar los elementos y los aspectos relevantes del proyecto, y los criterios de aceptación de los usuarios finales. El objetivo de esta actividad es obtener un panorama general del proyecto que permita conducir las siguientes actividades de la metodología Scrum-UIA.

##### 4.4.4.1. Tarea 1.4.1. Determinar los elementos y los aspectos relevantes del proyecto

**Descripción:** En esta tarea, se realizan reuniones entre el Arquitecto de la Información (L.R.) y el Dueño del Producto (J.P.) para analizar el contexto, el contenido y los usuarios finales del proyecto, con el fin de determinar los elementos y los aspectos relevantes del SI.

Para llevar a cabo la Tarea 1.4.1, primero se identificarán los aspectos relevantes del proyecto, que se pueden registrar, editar y consultar a través del módulo “Gestionar Aspectos (Mod-63)” de Scrum-UIA-MAT (ver sección 3.5.3.6). En la Figura 56, se muestra el registro de los aspectos relevantes del proyecto a través del soporte del módulo “Gestionar Aspectos (Mod-63)”. Como se puede observar en la Figura 56, los aspectos relevantes identificados para el SI de gestión de aprendizaje corresponden a: Usabilidad, Contenido y Valor de Negocio. Estos aspectos permiten considerar formalmente las preferencias relacionadas con los usuarios finales, la AI y el proyecto, respectivamente.

Relevant Aspects of the Project		
	Id	Name
<a href="#">Edit</a>	12	Usabilidad
<a href="#">Edit</a>	13	Contenido
<a href="#">Edit</a>	14	Valor de Negocio

id

name

[Insert](#) [Cancel](#)

Mod-63

All Rights Reserved - Contact Us

Universidad Autónoma de Madrid-Escuela Politécnica Superior

Search This Site

[Search](#)

Links

- Add End-Users to the Project
- Add Aspects to the Project
- Add Elements to Project's Aspects
- Set Up Project's Aspects Priority
- Manage Team Capacity

**Figura 56:** Ingreso y consulta de los aspectos mediante el uso del módulo “Gestionar Aspectos (Mod-63)”.

En el segundo y último paso requerido para completar la Tarea 1.4.1, se identificarán los elementos que componen a los aspectos relevantes definidos anteriormente. Este paso es soportado por el módulo “Gestionar Elementos (Mod-64)”, que permite registrar y

consultar todos los elementos de cada uno de los aspectos (ver sección 3.5.3.6). En la Figura 57, se presentan los elementos de los aspectos del SI de aprendizaje, gestionados por el módulo “Gestionar Elementos (Mod-64)”.

**Scrum-UIA-MAT**  
Scrum-UIA MAnagement Tool

Project | Product Backlog | Sprints | Task Board | Burndown | Set up | Project: Gestión de Aprendizaje

**Add Elements of the Aspects**

Project:

Aspects: ☒ Contenido ☐ Usabilidad ☐ Valor de Negocio

Name:

Description:

Priority:

Aspect	Name	Description	Priority
Content	Registro de las comunicaciones entre estudiantes		Low
Content	Descripción de las entidades de conocimiento		Medium
Content	Evaluaciones de los actividades de evaluación		Medium
Content	Feedback de las actividades de evaluación		Medium
Content	Registro de las actividades de los estudiantes		Medium

1 2 3

Mod-64

Search This Site:

**Links**

- Add End-Users to the Project
- Add Aspects to the Project
- Add Elements to Project's Aspects
- Set Up Project's Aspects Priority
- Manage Team Capacity

**Figura 57:** Ingreso y consulta de los elementos de los aspectos relevante del SI de aprendizaje, mediante el uso del módulo “Gestionar Elementos (Mod-64)”.

**Información de entrada:** Las principales fuentes de información provienen del Informe sobre las características principales del contexto (P1) de la Tarea 1.1.1, del Informe sobre las amenazas y oportunidades del contexto (P2) de la Tarea 1.1.2, del Informe sobre la identificación de los contenidos del proyecto (P3) de la Tarea 1.2.1, del Modelo de contenido general del proyecto (P4) de la Tarea 1.2.1, del Informe sobre el análisis de las prioridades del contenido (P5) de la Tarea 1.2.2, del Informe preliminar de los usuarios finales del proyecto y sus características (P6) de la Tarea 1.3.1 y del Informe final sobre las necesidades de los usuarios finales (P7) de la Tarea 1.3.2

**Producto de salida:** El producto de salida corresponde a un Informe final sobre los elementos y los aspectos relevantes del proyecto (P8). Este producto también se puede generar y consultar mediante el módulo “Gestionar Elementos (Mod-64)” (ver Figura 57). A continuación, se presenta el producto de salida:



**P8. Informe final sobre los elementos y los aspectos relevantes del proyecto**

**Autor:** Arquitecto de la Información (L.R.)

**Fecha:** 28 Julio 2016

<b>Identificador</b>	<b>Aspectos relevantes del SI de gestión de aprendizaje</b>	<b>Prioridad</b>
<b>12</b>	Usabilidad	Alta
<b>13</b>	Contenido	Media
<b>14</b>	Valor de Negocio	Baja

**Elementos del aspecto relevante de usabilidad**

**AE12.1:** Proveer adaptabilidad al entorno de aprendizaje.

**Prioridad:** Alta.

**AE12.2:** Proveer facilidad de aprendizaje.

**Prioridad:** Alta.

**AE12.3:** Proveer accesibilidad a la información (opciones de formato).

**Prioridad:** Alta.

**AE12.4:** Reducir la carga mental de trabajo.

**Prioridad:** Media.

**AE12.5:** Contemplar los potenciales errores.

**Prioridad:** Baja.

**AE12.6:** Proveer facilidad del SI para que éste sea recordado.

**Prioridad:** Alta.

**Elementos del aspecto relevante de contenido**

**AE13.1:** Datos personales de los estudiantes.

**Prioridad:** Media.

**AE13.2:** Contenido de las entidades de conocimiento.

**Prioridad:** Alta.

**AE13.3:** Datos personales de los instructores.

**Prioridad:** Baja.

**AE13.4:** Comunicaciones entre estudiantes e instructores.

**Prioridad:** Media.

**P8. Informe final sobre los elementos y los aspectos relevantes del proyecto**

**Autor:** Arquitecto de la Información (L.R.)

**Fecha:** 28 Julio 2016

**AE13.5:** Preferencias de los estudiantes.

**Prioridad:** Alta.

**AE13.6:** Descripción de las entidades de conocimiento.

**Prioridad:** Alta.

**AE13.7:** Retroalimentación de las actividades de evaluación.

**Prioridad:** Alta.

**AE13.8:** Registro de las actividades de los estudiantes.

**Prioridad:** Alta.

**Elementos del aspecto relevante de valor de negocio**

**AE14.1:** Gestionar el progreso de los estudiantes.

**Prioridad:** Medio.

**AE14.2:** Gestionar la matricula.

**Prioridad:** Alta.

**AE14.3:** Monitorear el proceso de aprendizaje.

**Prioridad:** Alta.

**AE14.4:** Gestionar las entidades de conocimiento.

**Prioridad:** Alta.

**AE14.5:** Organizar el almacenamiento de la información.

**Prioridad:** Media.

**AE14.6:** Coordinar composición de las entidades de conocimiento.

**Prioridad:** Alta.

**Producto de Verificación 8:** Informe final sobre los elementos y los aspectos relevantes del proyecto.

**4.4.4.2. Tarea 1.4.2. Determinar criterios de aceptación de los usuarios finales**

**Descripción:** El objetivo de esta tarea es definir los criterios de aceptación de los usuarios finales, identificados en el Informe preliminar de los usuarios finales del proyecto y sus características (P6), de la Tarea 3.1.1.

**Información de entrada:** Las fuentes principales de información son el Informe sobre las características principales del contexto (P1) de la Tarea 1.1.1, el Informe sobre las amenazas y oportunidades del contexto (P2) de la Tarea 1.1.2, el Informe preliminar de los usuarios finales del proyecto y sus características (P6) de la Tarea 1.3.1, y el Informe final sobre las necesidades de los usuarios finales (P7) de la Tarea 1.3.2.

**Producto de salida:** A continuación, se presenta el Catálogo de criterios de aceptación de los usuarios finales (P9) que se generó en esta tarea:

<b>P9. Catálogo de criterios de aceptación de los usuarios finales</b>	
<b>Autor:</b> Arquitecto de la Información (L.R.)	
<b>Fecha:</b> 28 Julio 2016	
<b>CAUF31.1:</b> Comprobar las acciones.	
<b>Usuario Final:</b> Estudiante.	
<b>Descripción:</b> Permitir comprobar cómo las acciones afectan a la salida del SI.	
<b>CAUF31.2:</b> Interfaz adaptable.	
<b>Usuario Final:</b> Estudiante.	
<b>Descripción:</b> Mostrar opciones para personalizar la interfaz, de acuerdo al perfil del usuario.	
<b>CAUF31.3:</b> Presentación de la información.	
<b>Usuario Final:</b> Estudiante.	
<b>Descripción:</b> Mostrar alternativas gráficas en la presentación de la información.	
<b>CAUF32.1:</b> Acciones reversibles.	
<b>Usuario Final:</b> Instructor.	
<b>Descripción:</b> Posibilitar la reversibilidad y la recuperabilidad de las acciones.	
<b>CAUF33.1:</b> Gestión de errores.	
<b>Usuario Final:</b> Administrador.	
<b>Descripción:</b> Posibilitar la recuperación de los errores.	
<b>CAUF34.1:</b> Alternativas de accesos.	
<b>Usuario Final:</b> Observador.	
<b>Descripción:</b> Ofrecer alternativas para acceder al SI.	

**Producto de Verificación 9:** Catálogo de criterios de aceptación de los usuarios finales.

#### 4.4.5. Actividad 2.1. Identificación y análisis de requisitos del proyecto

Una vez configurado el panorama general del proyecto, en esta actividad se identificarán y analizarán los requisitos del SI de gestión de aprendizaje de la universidad.

##### 4.4.5.1. Tarea 2.1.1. Identificar requisitos del proyecto

**Descripción:** En esta tarea se recopilan los requisitos del SI de gestión de aprendizaje de la universidad. El objetivo es recabar todos los requisitos del proyecto, considerando las prioridades identificadas en las actividades anteriores.

**Información de entrada:** La principal información de entrada de esta tarea corresponde a los requisitos funcionales del proyecto (E5). Esta información se complementa con las fuentes sobre el Informe final sobre los elementos y los aspectos relevantes del proyecto (P8) de la Tarea 1.4.1 y el Catálogo de criterios de aceptación de los usuarios finales (P9) de la Tarea 1.4.2. Todas estas fuentes de información se deberán analizar en conjunto, entre el Dueño del Producto y el Arquitecto de la Información, mediante técnicas tales como reuniones, entrevistas no estructuradas y cuestionarios.

**Producto de salida:** A continuación, se presenta el Informe preliminar de requisitos del proyecto (P10) que se generó en esta tarea:

P10. Informe preliminar de requisitos del proyecto	
<b>Autor:</b> Dueño del Producto (J.P.)	
<b>Fecha:</b> 28 Julio 2016	
<b>R1:</b> Crear y editar cursos	
<b>Descripción:</b> Crear y editar los cursos ofrecidos por la universidad, con el fin de configurar sus datos generales (por ejemplo, nombre, objetivos, requisitos, profesores).	
<b>R2:</b> Gestionar cursos en un entorno integrado	
<b>Descripción:</b> Proveer un entorno virtual que permite consultar y gestionar los diferentes cursos que se llevan a cabo.	
<b>R3:</b> Crear y editar actividades de evaluación	
<b>Descripción:</b> Generar actividades de evaluación, asociadas a los cursos impartidos por la universidad, con el fin de permitir su gestión electrónica en todas sus fases.	
<b>R4:</b> Configurar perfiles de estudiantes y profesores	
<b>Descripción:</b> Configurar datos personales y preferencias de los estudiantes y profesores.	
<b>R5:</b> Gestionar las inscripciones en los cursos	

<b>P10. Informe preliminar de requisitos del proyecto</b>	
<b>Autor:</b> Dueño del Producto (J.P.)	
<b>Fecha:</b> 28 Julio 2016	
<b>Descripción:</b>	Proveer un entorno integrado que permita conducir el proceso de matrícula de los estudiantes.
<b>R6:</b>	Incorporar material en los cursos
<b>Descripción:</b>	Enlazar y proveer documentos en diferentes formatos, asociados a los cursos impartidos por la universidad.
<b>R7:</b>	Consultar los eventos y tareas de los cursos
<b>Descripción:</b>	Proveer una estructura que permita consultar y navegar a través de los eventos y las tareas de los cursos.
<b>R8:</b>	Consultar información sobre el progreso de los estudiantes
<b>Descripción:</b>	Generar informes sobre las interacciones de los estudiantes en los cursos y las estadísticas de uso del SI.
<b>R9:</b>	Consultar evaluaciones de las actividades
<b>Descripción:</b>	Permitir revisar los resultados de las actividades de evaluación y notificar sobre los ciclos requeridos.
<b>R10:</b>	Enviar, de manera electrónica, las actividades de evaluación
<b>Descripción:</b>	Permitir adjuntar y relacionar documentos digitales con las actividades de evaluación.
<b>R11:</b>	Crear entornos de discusión
<b>Descripción:</b>	Proveer grupo de discusión entre los integrantes de los cursos.
<b>R12:</b>	Proveer un repositorio de archivos
<b>Descripción:</b>	Proveer un espacio virtual centralizado para almacenar y mantener información digital.
<b>R13:</b>	Crear aulas virtuales
<b>Descripción:</b>	Proveer entorno virtual que permita realizar enseñanza en línea.
<b>R14:</b>	Revisar el SI como otro usuario
<b>Descripción:</b>	Permitir simular el SI, de acuerdo a diferentes perfiles y usuarios.

**Producto de Verificación 10:** Informe preliminar de requisitos del proyecto.

#### 4.4.5.2. Tarea 2.1.2. Analizar requisitos del proyecto

**Descripción:** Esta tarea analiza los requisitos, identificados en el Informe preliminar de requisitos del proyecto (P10) de la tarea anterior (Tarea 2.1.1).

**Información de entrada:** Las principales fuentes de información corresponden al Informe final sobre los elementos y los aspectos relevantes del proyecto (P8) de la Tarea 1.4.1, al Catálogo de criterios de aceptación de los usuarios finales (P9) de la Tarea 1.4.2 y al Informe preliminar de requisitos del proyecto (P10) de la Tarea 2.1.1. Todas estas fuentes se deberán examinar entre el Dueño del Producto y el Arquitecto de la Información mediante las reuniones.

**Producto de salida:** A continuación, se presenta el producto (P11) producido en esta tarea. Este producto analiza los requisitos, considerando la aportación entregada al valor de negocio.

P11. Catálogo de requisitos analizados del proyecto	
<b>Autor:</b> Dueño del Producto (J.P.)	
<b>Fecha:</b> 28 Julio 2016	
<b>R1:</b> Crear y editar cursos.	
<b>Descripción:</b> Crear y editar los cursos ofrecidos por la universidad, con el fin de configurar sus datos generales (por ejemplo, nombre, objetivos, requisitos, profesores).	
<b>Usuarios finales:</b> Instructores, Administradores.	
<b>Análisis:</b> Este requisito permite definir y configurar las materias que forman los planes de estudio de la universidad.	
<b>R2:</b> Gestionar cursos en un entorno integrado.	
<b>Descripción:</b> Proveer un entorno virtual que permite consultar y gestionar los diferentes cursos que se llevan a cabo.	
<b>Usuarios finales:</b> Estudiantes.	
<b>Análisis:</b> Este requisito permite que la información de aprendizaje de los estudiantes se pueda organizar y configurar.	
<b>R3:</b> Crear y editar actividades de evaluación.	
<b>Descripción:</b> Generar actividades de evaluación, asociadas a los cursos impartidos por la universidad, con el fin de permitir su gestión electrónica en todas sus fases.	
<b>Usuarios finales:</b> Instructores.	
<b>Análisis:</b> Este requisito permite digitalizar las instancias de evaluación.	

<b>P11. Catálogo de requisitos analizados del proyecto</b> <b>Autor:</b> Dueño del Producto (J.P.) <b>Fecha:</b> 28 Julio 2016	
<b>R4:</b> Configurar perfiles de estudiantes y profesores. <b>Descripción:</b> Configurar datos personales y preferencias de los estudiantes y de los profesores. <b>Usuarios finales:</b> Administradores. <b>Análisis:</b> Este requisito permite digitalizar los perfiles de los estudiantes y profesores, y enlazarlos con otros entornos digitales.	
<b>R5:</b> Gestionar las inscripciones en los cursos. <b>Descripción:</b> Proveer un entorno integrado que permite conducir el proceso de matrícula de los estudiantes. <b>Usuarios finales:</b> Administradores. <b>Análisis:</b> Este requisito facilita que la composición de los cursos se pueda tramitar.	
<b>R6:</b> Incorporar material en los cursos. <b>Descripción:</b> Enlazar y proveer documentos en diferentes formatos, asociados a los cursos impartidos por la universidad. <b>Usuarios finales:</b> Instructores. <b>Análisis:</b> Este requisito permite definir los contenidos de los cursos.	
<b>R7:</b> Consultar los eventos y tareas de los cursos. <b>Descripción:</b> Proveer una estructura que permita consultar y navegar a través de los eventos y las tareas de los cursos. <b>Usuarios finales:</b> Estudiantes, Observadores. <b>Análisis:</b> Este requisito provee información de los cursos a los estudiantes.	
<b>R8:</b> Consultar información sobre el progreso de los estudiantes. <b>Descripción:</b> Generar informes sobre las interacciones de los estudiantes en los cursos y las estadísticas de uso del SI. <b>Usuarios finales:</b> Instructores, Observadores. <b>Análisis:</b> Este requisito permite monitorear el proceso de aprendizaje de los estudiantes.	
<b>R9:</b> Consultar evaluaciones de las actividades.	

<b>P11. Catálogo de requisitos analizados del proyecto</b>	
<b>Autor:</b> Dueño del Producto (J.P.)	
<b>Fecha:</b> 28 Julio 2016	
<p><b>Descripción:</b> Revisar los resultados de las actividades de evaluación y notificar sobre los ciclos requeridos.</p> <p><b>Usuarios finales:</b> Estudiantes, Observadores.</p> <p><b>Análisis:</b> Este requisito provee retroalimentación a los estudiantes respecto a los trabajos de evaluación realizados.</p>	
<p><b>R10:</b> Enviar, de manera electrónica, las actividades de evaluación.</p> <p><b>Descripción:</b> Permitir adjuntar y relacionar documentos digitales con las actividades de evaluación.</p> <p><b>Usuarios finales:</b> Estudiantes.</p> <p><b>Análisis:</b> Este requisito facilita acceder a los trabajos de evaluación de los estudiantes y proporciona un registro automático del proceso.</p>	
<p><b>R11:</b> Crear entornos de discusión.</p> <p><b>Descripción:</b> Proveer Chat y grupo de discusión entre los integrantes de los cursos.</p> <p><b>Usuarios finales:</b> Instructores.</p> <p><b>Análisis:</b> Este requisito permite promover la interacción y el trabajo en equipo de los estudiantes.</p>	
<p><b>R12:</b> Proveer un repositorio de archivos.</p> <p><b>Descripción:</b> Proveer un espacio virtual centralizado para almacenar y mantener información digital.</p> <p><b>Usuarios finales:</b> Estudiantes.</p> <p><b>Análisis:</b> Este requisito provee una capacidad de almacenamiento virtual para que los estudiantes puedan centralizar y gestionar los recursos.</p>	
<p><b>R13:</b> Crear aulas virtuales.</p> <p><b>Descripción:</b> Proveer entorno virtual que permita realizar enseñanza en línea.</p> <p><b>Usuarios finales:</b> Instructores.</p> <p><b>Análisis:</b> Este requisito permite a los estudiantes acceder al material de estudio y, al mismo tiempo, interactuar con el profesor y con otros estudiantes.</p>	
<b>R14:</b> Revisar el SI como otro usuario.	



### **P11. Catálogo de requisitos analizados del proyecto**

**Autor:** Dueño del Producto (J.P.)

**Fecha:** 28 Julio 2016

**Descripción:** Permitir simular el SI, de acuerdo a diferentes perfiles.

**Usuarios finales:** Administrador.

**Análisis:** Este requisito permite verificar las funcionalidades del SI.

**Producto de Verificación 11:** Catálogo de requisitos analizados del proyecto.

#### **4.4.6. Actividad 2.2. Generación final de requisitos**

Una vez identificados los requisitos del SI de gestión de aprendizaje de la universidad, en esta actividad se realizará su generación final. Esto consiste en configurar cada requisito en formato de historias de usuarios, estimarlos en puntos de historia (y días ideales) y priorizarlos de acuerdo a los elementos y aspectos relevantes del proyecto (utilizando el enfoque del método QMSPR, descrito en la sección 3.3).

##### **4.4.6.1. Tarea 2.2.1. Análisis y resolución de conflicto de requisitos**

**Descripción:** El objetivo de esta tarea es analizar las dependencias y conflictos de los requisitos.

**Información de entrada:** Las principales fuentes de información corresponden al Informe final sobre los elementos y los aspectos relevantes del proyecto (P8) de la Tarea 1.4.1, al Catálogo de criterios de aceptación de los usuarios finales (P9) de la Tarea 1.4.2, al Informe preliminar de requisitos del proyecto (P10) de la Tarea 2.1.1, y al Catálogo de requisitos analizados del proyecto (P11) de la Tarea 2.1.2.

**Producto de salida:** El principal resultado es el Catálogo final revisado de requisitos del proyecto (P12) que permite identificar las dependencias de los requisitos. Este producto se presenta a continuación:

### **P12. Catálogo final revisado de requisitos del proyecto**

**Autor:** Dueño del Producto (J.P.)

**Fecha:** 28 Julio 2016

**R1:** Crear y editar cursos.

**Descripción:** Crear y editar los cursos ofrecidos por la universidad, con el fin de configurar sus datos generales (por ejemplo, nombre, objetivos, requisitos, profesores).

**Usuarios finales:** Instructores, Administradores.

## P12. Catálogo final revisado de requisitos del proyecto

**Autor:** Dueño del Producto (J.P.)

**Fecha:** 28 Julio 2016

**Análisis:** Este requisito permite definir y configurar las materias que forman los planes de estudio de la universidad.

**R2:** Gestionar cursos en un entorno integrado.

**Descripción:** Proveer un entorno virtual, que permite consultar y gestionar los diferentes cursos que se llevan a cabo.

**Usuarios finales:** Estudiantes.

**Análisis:** Este requisito permite que la información de aprendizaje de los estudiantes se pueda organizar y configurar.

**Dependencias:** R1.

**R3:** Crear y editar actividades de evaluación.

**Descripción:** Generar actividades de evaluación, asociadas a los cursos impartidos por la universidad, con el fin de permitir su gestión electrónica en todas sus fases.

**Usuarios finales:** Instructores.

**Análisis:** Este requisito permite digitalizar las instancias de evaluación.

**Dependencias:** R1.

**R4:** Configurar perfiles de estudiantes y profesores.

**Descripción:** Configurar datos personales y preferencias de los estudiantes y profesores.

**Usuarios finales:** Administradores.

**Análisis:** Este requisito permite digitalizar los perfiles de los estudiantes y profesores, y enlazarlos con otros entornos digitales.

**Dependencias:** R1.

**R5:** Gestionar las inscripciones en los cursos.

**Descripción:** Proveer un entorno integrado que permite conducir el proceso de matrícula de los estudiantes.

**Usuarios finales:** Administradores.

**Análisis:** Este requisito facilita que la composición de los cursos se pueda tramitar.

**Dependencias:** R1, R2 y R4.

## P12. Catálogo final revisado de requisitos del proyecto

**Autor:** Dueño del Producto (J.P.)

**Fecha:** 28 Julio 2016

**R6:** Incorporar material en los cursos.

**Descripción:** Enlazar y proveer documentos en diferentes formatos, asociados a los cursos impartidos por la universidad.

**Usuarios finales:** Instructores.

**Análisis:** Este requisito permite definir los contenidos de los cursos.

**Dependencias:** R1 y R2.

**R7:** Consultar los eventos y tareas de los cursos.

**Descripción:** Proveer una estructura que permita consultar y navegar a través de los eventos y las tareas de los cursos.

**Usuarios finales:** Estudiantes, Observadores.

**Análisis:** Este requisito provee información de los cursos a los estudiantes.

**Dependencias:** R1 y R2.

**R8:** Consultar información sobre el progreso de los estudiantes.

**Descripción:** Generar informes sobre las interacciones de los estudiantes en los cursos y las estadísticas de uso del SI.

**Usuarios finales:** Instructores, Observadores.

**Análisis:** Este requisito permite monitorear el proceso de aprendizaje de los estudiantes.

**Dependencias:** R1, R2 y R4.

**R9:** Consultar evaluaciones de las actividades.

**Descripción:** Revisar los resultados de las actividades de evaluación y notificar sobre los ciclos requeridos.

**Usuarios finales:** Estudiantes, Observadores.

**Análisis:** Este requisito provee retroalimentación a los estudiantes respecto a los trabajos de evaluación realizados.

**Dependencias:** R1, R2 y R3.

**R10:** Enviar, de manera electrónica, las actividades de evaluación.

**Descripción:** Permitir adjuntar y relacionar documentos digitales con las actividades

<b>P12. Catálogo final revisado de requisitos del proyecto</b> <b>Autor:</b> Dueño del Producto (J.P.) <b>Fecha:</b> 28 Julio 2016	
de evaluación.	<b>Usuarios finales:</b> Estudiantes. <b>Análisis:</b> Este requisito facilita acceder a los trabajos de evaluación de los estudiantes y proporciona un registro automático del proceso. <b>Dependencias:</b> R1, R2 y R3.
<b>R11:</b> Crear entornos de discusión. <b>Descripción:</b> Proveer Chat y grupo de discusión entre los integrantes de los cursos. <b>Usuarios finales:</b> Instructores. <b>Análisis:</b> Este requisito permite promover la interacción y el trabajo en equipo de los estudiantes. <b>Dependencias:</b> R1, R2 y R4.	
<b>R12:</b> Proveer un repositorio de archivos. <b>Descripción:</b> Proveer un espacio virtual centralizado para almacenar y mantener información digital. <b>Usuarios finales:</b> Estudiantes. <b>Análisis:</b> Este requisito provee una capacidad de almacenamiento virtual para que los estudiantes puedan centralizar y gestionar recursos. <b>Dependencias:</b> R1.	
<b>R13:</b> Crear aulas virtuales. <b>Descripción:</b> Proveer entorno virtual que permita realizar enseñanza en línea. <b>Usuarios finales:</b> Instructores. <b>Análisis:</b> Este requisito permite a los estudiantes acceder al material de estudio y, al mismo tiempo, interactuar con el profesor y con otros estudiantes. <b>Dependencias:</b> R1 y R2.	
<b>R14:</b> Revisar el SI como otro usuario. <b>Descripción:</b> Simular el SI, de acuerdo a diferentes perfiles. <b>Usuarios finales:</b> Administrador. <b>Análisis:</b> Este requisito permite verificar las funcionalidades del SI.	

**P12. Catálogo final revisado de requisitos del proyecto**

**Autor:** Dueño del Producto (J.P.)

**Fecha:** 28 Julio 2016

**Dependencias:** R1, R2 y R4.

**Producto de Verificación 12:** Catálogo final revisado de requisitos del proyecto.

**4.4.6.2. Tarea 2.2.2. Generación de requisitos**

**Descripción:** El objetivo de esta tarea es configurar los requisitos identificados y analizados (Tarea 2.2.1) del SI de gestión de aprendizaje de la universidad, mediante una Lista del Producto con sus elementos descritos en formato de historias de usuario.

En la Tarea 2.2.2, los requisitos se configurarán estableciendo los siguientes datos: el nombre en formato de historia de usuario, la descripción, la clasificación, los criterios de aceptación de los usuarios finales y las dependencias. Asimismo, en la realización de la Tarea 2.2.2, se utilizará el módulo “Gestionar Historias de usuarios (Mod-22)” de Scrum-UIA-MAT (ver sección 3.5.3.2). Este módulo permite registrar, editar, eliminar y consultar los requisitos.

En la Figura 58, se muestra el registro del requisito Gestionar cursos en un entorno integrado (R2) a través del módulo “Gestionar Historias de usuarios (Mod-22)”. Como se puede observar en la Figura 58, el módulo permite registrar el requisito en formato de historias de usuario, identificando los usuarios finales, el objetivo y el valor de negocio del requisito (ver campo Name del módulo en la Figura 58). Asimismo, el módulo permite especificar la descripción y la clasificación del requisito a través de los campos Descripción y Type, respectivamente. Finalmente, las dependencias y los criterios de aceptación de los usuarios finales se especifican en los apartados Dependencies y End-User Acceptance Criteria del módulo, respectivamente.

Cabe destacar que el requisito anterior corresponde a una Historia de Usuario Usable de la AI (ver la clasificación del requisito en la Figura 58, es decir, campo Type). Esta clasificación del requisito permite identificar a las historias de usuarios que están relacionadas con la AI, y facilita conducir el desarrollo incremental a través de los niveles de fidelidad de los entregables de la AI (en la sección 3.4 se describió un ejemplo para identificar estos requisitos).

Backlog Item	
Project:	Gestión de Aprendizaje
Name:	*As a Estudiante, I want un entorno virtual so that la información de
Description:	Proveer un entorno virtual que permite consultar y gestionar los diferentes cursos que se llevan a cabo.
Rough Estimate:	*15 story points
Type:	*IAU User Story
Estimate:	*5 Days
Remaining:	*5 Days
Status:	New
Sprint:	Select Sprint
Priority	
Product Owner:	Set Up
Information Architect:	Set Up
Related Aspects	
Usability:	Proveer adaptabilidad del entorno de aprendizaje
Contents:	Descripción de las entidades de conocimiento
Business Value:	Gestionar el progreso de los estudiantes
Dependencies	
User Story:	1 selected
End-User Acceptance Criteria	
<ul style="list-style-type: none"> <li>•CAUF31.2: Interfaz adaptable.</li> <li>•CAUF31.3: Presentación de la información.</li> </ul>	
Mod-22	Delete Item Modify Item Cancel

**Figura 58:** Registro de los requisitos en formato de historias de usuarios a través del módulo “Gestionar Historias de usuarios (Mod-22)”.

**Información de entrada:** Las principales fuentes de información de esta tarea corresponden al Informe final sobre los elementos y los aspectos relevantes del proyecto (P8) de la Tarea 1.4.1, al Catálogo de criterios de aceptación de los usuarios finales (P9) de la Tarea 1.4.2, al Informe preliminar de requisitos del proyecto (P10) de la Tarea 2.1.1, al Catálogo de requisitos analizados del proyecto (P11) de la Tarea 2.1.2 y al Catálogo final revisado de requisitos del proyecto (P12) de la Tarea 2.2.1.

**Producto de salida:** El principal resultado es la Lista del Producto con los elementos descritos en formato de Historia de Usuario (P13) que permite identificar la descripción, la clasificación, los criterios de aceptación de los usuarios finales y las dependencias de los requisitos. Asimismo, los datos de este producto se pueden generar y consultar mediante el módulo “Gestionar Historias de usuarios (Mod-22)” (ver Figura 58), y organizar a través del módulo “Consultar Historias de usuarios (Mod-21)” (en la sección 3.5.3.6, se

encuentran los detalles de estos módulos). Los detalles de este producto (P13) se presentan a continuación:

**P13. Lista del Producto con los elementos descritos en formato de Historia de Usuario**

**Autor:** Dueño del Producto (J.P.)

**Fecha:** 29 Julio 2016

**R1:** Crear y editar cursos.

**Como** Instructores y Administradores, **Queremos** crear y editar los cursos, **con el fin de** definir y configurar las materias que forman los planes de estudio de la universidad.

**Descripción:** Crear y editar los cursos ofrecidos por la universidad, con el fin de configurar sus datos generales (por ejemplo, nombre, objetivos, requisitos, profesores).

**Criterios de aceptación de los usuarios finales:**

- **CAUF32.1:** Acciones reversibles.
- **CAUF33.1:** Gestión de errores.

**R2:** Gestionar cursos en un entorno integrado.

**Como** Estudiante, **Quiero** un entorno virtual, **con el fin de** que la información de aprendizaje se pueda organizar y configurar.

**Descripción:** Proveer un entorno virtual que permita consultar y gestionar los diferentes cursos que se llevan a cabo.

**Dependencias:** R1.

**Criterios de aceptación de los usuarios finales:**

- **CAUF31.2:** Interfaz adaptable.
- **CAUF31.3:** Presentación de la información.

**R3:** Crear y editar actividades de evaluación.

**Como** Instructor, **Quiero** generar actividades de evaluación, **con el fin de** digitalizar las instancias de evaluación.

**Descripción:** Generar actividades de evaluación asociadas a los cursos impartidos por la universidad, con el fin de permitir su gestión electrónica en todas sus fases.

**Dependencias:** R1.

<p><b>P13. Lista del Producto con los elementos descritos en formato de Historia de Usuario</b></p> <p><b>Autor:</b> Dueño del Producto (J.P.)</p> <p><b>Fecha:</b> 29 Julio 2016</p>
<p><b>Criterios de aceptación de los usuarios finales:</b></p> <ul style="list-style-type: none"> <li>• <b>CAUF32.1:</b> Acciones reversibles.</li> </ul>
<p><b>R4:</b> Configurar perfiles de estudiantes y profesores.</p> <p><b>Como</b> Administrador, <b>Quiero</b> configurar preferencias de los estudiantes y profesores, <b>con el fin de</b> digitalizar los perfiles y enlazarlos con otros entornos digitales.</p> <p><b>Descripción:</b> Configurar datos personales y preferencias de los estudiantes y profesores.</p> <p><b>Dependencias:</b> R1.</p> <p><b>Criterios de aceptación de los usuarios finales:</b></p> <ul style="list-style-type: none"> <li>• <b>CAUF33.1:</b> Gestión de errores.</li> </ul>
<p><b>R5:</b> Gestionar las inscripciones en los cursos.</p> <p><b>Como</b> Administrador, <b>Quiero</b> registrar y consultar las inscripciones de los cursos, <b>con el fin de</b> tramitar su composición.</p> <p><b>Descripción:</b> Proveer un entorno integrado que permita conducir el proceso de matrícula de los estudiantes.</p> <p><b>Dependencias:</b> R1, R2 y R4.</p> <p><b>Criterios de aceptación de los usuarios finales:</b></p> <ul style="list-style-type: none"> <li>• <b>CAUF33.1:</b> Gestión de errores.</li> </ul>
<p><b>R6:</b> Incorporar material en los cursos.</p> <p><b>Como</b> Instructor, <b>Quiero</b> proveer documentos en los cursos, <b>con el fin de</b> definir sus contenidos.</p> <p><b>Descripción:</b> Enlazar y proveer documentos en diferentes formatos, asociados a los cursos impartidos por la universidad.</p> <p><b>Dependencias:</b> R1 y R2.</p> <p><b>Criterios de aceptación de los usuarios finales:</b></p> <ul style="list-style-type: none"> <li>• <b>CAUF32.1:</b> Acciones reversibles.</li> </ul>
<p><b>R7:</b> Consultar los eventos y tareas de los cursos.</p>



### **P13. Lista del Producto con los elementos descritos en formato de Historia de Usuario**

**Autor:** Dueño del Producto (J.P.)

**Fecha:** 29 Julio 2016

**Como** estudiante y observador, **Queremos** consultar los eventos y tareas de los cursos, **con el fin de** proveer información a los estudiantes.

**Descripción:** Proveer una estructura que permita consultar y navegar a través de los eventos y las tareas de los cursos.

**Dependencias:** R1 y R2.

**Criterios de aceptación de los usuarios finales:**

- **CAUF31.3:** Presentación de la información.
- **CAUF34.1:** Alternativas de accesos.

**R8:** Consultar información sobre el progreso de los estudiantes.

**Como** Instructor y observador, **Queremos** revisar el progreso de los estudiantes, **con el fin de** monitorear su proceso de aprendizaje.

**Descripción:** Generar informes sobre las interacciones de los estudiantes en los cursos y las estadísticas de uso del SI.

**Dependencias:** R1, R2 y R4.

**Criterios de aceptación de los usuarios finales:**

- **CAUF32.1:** Acciones reversibles.
- **CAUF34.1:** Alternativas de accesos.

**R9:** Consultar evaluaciones de las actividades.

**Como** Estudiante y observador, **Queremos** revisar las evaluaciones de las actividades, **con el fin de** obtener retroalimentación.

**Descripción:** Permitir revisar los resultados de las actividades de evaluación y notificar sobre los ciclos requeridos.

**Dependencias:** R1, R2 y R3.

**Criterios de aceptación de los usuarios finales:**

- **CAUF31.2:** Interfaz adaptable.
- **CAUF31.3:** Presentación de la información.
- **CAUF34.1:** Alternativas de accesos.

### **P13. Lista del Producto con los elementos descritos en formato de Historia de Usuario**

**Autor:** Dueño del Producto (J.P.)

**Fecha:** 29 Julio 2016

**R10:** Enviar, de manera electrónica, las actividades de evaluación.

**Como** Estudiante, **Quiero** enviar actividades de evaluación, **con el fin de** proporcionar un registro automático del proceso.

**Descripción:** Permitir adjuntar y relacionar documentos digitales con las actividades de evaluación.

**Dependencias:** R1, R2 y R3.

**Criterios de aceptación de los usuarios finales:**

- **CAUF31.1:** Comprobar las acciones
- **CAUF31.2:** Interfaz adaptable.

**R11:** Crear entornos de discusión.

**Como** Instructor, **Quiero** crear instancias de discusión, **con el fin de** promover la interacción y el trabajo en equipo de los estudiantes.

**Descripción:** Proveer Chat y grupo de discusión entre los integrantes de los cursos.

**Dependencias:** R1, R2 y R4.

**Criterios de aceptación de los usuarios finales:**

- **CAUF32.1:** Acciones reversibles.

**R12:** Proveer un repositorio de archivos.

**Como** Estudiante, **Quiero** disponer de un repositorio de archivo, **con el fin de** centralizar el almacenamiento virtual de los recursos.

**Descripción:** Proveer un espacio virtual centralizado para almacenar y mantener información digital.

**Dependencias:** R1.

**Criterios de aceptación de los usuarios finales:**

- **CAUF31.2:** Interfaz adaptable.
- **CAUF31.3:** Presentación de la información.

**R13:** Crear aulas virtuales.

**Como** Instructor, **Quiero** crear aulas virtuales, **con el fin de** proveer entorno virtual

<b>P13. Lista del Producto con los elementos descritos en formato de Historia de Usuario</b>	
<b>Autor:</b> Dueño del Producto (J.P.) <b>Fecha:</b> 29 Julio 2016	
para acceder al material de estudio e interactuar con los estudiantes. <b>Descripción:</b> Proveer entorno virtual que permita realizar enseñanza en línea. <b>Dependencias:</b> R1 y R2. <b>Criterios de aceptación de los usuarios finales:</b> <ul style="list-style-type: none"> <li>• <b>CAUF32.1:</b> Acciones reversibles.</li> </ul>	
<b>R14:</b> Revisar el SI como otro usuario. <b>Como</b> Administrador, <b>Quiero</b> simular el SI con el perfil de otro usuario, <b>con el fin de</b> verificar sus funcionalidades. <b>Descripción:</b> Permitir simular el sistema, de acuerdo a diferentes perfiles del SI. <b>Dependencias:</b> R1, R2 y R4. <b>Criterios de aceptación de los usuarios finales:</b> <ul style="list-style-type: none"> <li>• <b>CAUF33.1:</b> Gestión de errores.</li> </ul>	

**Producto de Verificación 13:** Lista del Producto con los elementos descritos en formato de Historia de Usuario.

#### 4.4.6.3. Tarea 2.2.3. Estimar el tamaño de los requisitos

**Descripción:** El objetivo de esta tarea es estimar el tamaño de los requisitos de la Lista del Producto.

En la ejecución de la Tarea 2.2.3, se utilizará el módulo “Gestionar Historias de usuarios (Mod-22)” (ver Figura 58). Este módulo permite registrar la estimación del tamaño de los requisitos en puntos de historia y días ideales (días, horas o minutos).

En la Figura 58, se muestra la estimación del tamaño del requisito Gestionar cursos en un entorno integrado (R2), a través del módulo “Gestionar Historias de usuarios (Mod-22)”. Como se puede observar, este módulo permite especificar la estimación en puntos de historias mediante el campo Rough Estimate. Asimismo, la estimación del esfuerzo total y del restante se especifican mediante los campos Estimate y Remaining, respectivamente.

**Información de entrada:** La Tarea 2.2.3 utiliza como fuente principal de información la Lista del Producto con los elementos descritos en formato de Historia de Usuario (P13), proporcionada por la Tarea 2.2.2.

**Producto de salida:** El principal resultado de la Tarea 2.2.3 corresponde al Informe de la estimación de los elementos de la Lista del Producto (P14). Al igual que en la Tarea 2.2.2, los datos de este producto también se pueden generar y consultar mediante el módulo “Gestionar Historias de usuarios (Mod-22)” (ver Figura 58), y organizar en el módulo “Consultar Historias de usuarios (Mod-21)”. Los detalles de este producto (P14) se presentan a continuación:

<b>P14. Informe de la estimación de los elementos de la Lista del Producto</b>	
<b>Autor:</b> Equipo de Desarrollo (M.R. y A.G.)	
<b>Fecha:</b> 29 Julio 2016	
<b>R1:</b> Crear y editar cursos	
<b>Puntos de historias:</b> 6	
<b>Estimación total:</b> 2 días	
<b>Estimación restante:</b> 2 días	
<b>R2:</b> Gestionar cursos en un entorno integrado	
<b>Puntos de historias:</b> 15	
<b>Estimación total:</b> 5 días	
<b>Estimación restante:</b> 5 días	
<b>R3:</b> Crear y editar actividades de evaluación	
<b>Puntos de historias:</b> 9	
<b>Estimación total:</b> 3 días	
<b>Estimación restante:</b> 3 días	
<b>R4:</b> Configurar perfiles de estudiantes y profesores	
<b>Puntos de historias:</b> 12	
<b>Estimación total:</b> 4 días	
<b>Estimación restante:</b> 4 días	
<b>R5:</b> Gestionar las inscripciones en los cursos	
<b>Puntos de historias:</b> 9	
<b>Estimación total:</b> 3 días	
<b>Estimación restante:</b> 3 días	
<b>R6:</b> Incorporar material en los cursos	

<p><b>P14. Informe de la estimación de los elementos de la Lista del Producto</b></p> <p><b>Autor:</b> Equipo de Desarrollo (M.R. y A.G.)</p> <p><b>Fecha:</b> 29 Julio 2016</p>
<p><b>Puntos de historias:</b> 3</p> <p><b>Estimación total:</b> 1 día</p> <p><b>Estimación restante:</b> 1 día</p>
<p><b>R7:</b> Consultar los eventos y tareas de los cursos</p> <p><b>Puntos de historias:</b> 6</p> <p><b>Estimación total:</b> 2 días</p> <p><b>Estimación restante:</b> 2 días</p>
<p><b>R8:</b> Consultar información sobre el progreso de los estudiantes</p> <p><b>Puntos de historias:</b> 9</p> <p><b>Estimación total:</b> 3 días</p> <p><b>Estimación restante:</b> 3 días</p>
<p><b>R9:</b> Consultar evaluaciones de las actividades</p> <p><b>Puntos de historias:</b> 3</p> <p><b>Estimación total:</b> 1 día</p> <p><b>Estimación restante:</b> 1 día</p>
<p><b>R10:</b> Enviar, de manera electrónica, las actividades de evaluación</p> <p><b>Puntos de historias:</b> 6</p> <p><b>Estimación total:</b> 2 días</p> <p><b>Estimación restante:</b> 2 días</p>
<p><b>R11:</b> Crear entornos de discusión</p> <p><b>Puntos de historias:</b> 9</p> <p><b>Estimación total:</b> 3 días</p> <p><b>Estimación restante:</b> 3 días</p>
<p><b>R12:</b> Proveer un repositorio de archivos</p> <p><b>Puntos de historias:</b> 12</p> <p><b>Estimación total:</b> 4 días</p>

<b>P14. Informe de la estimación de los elementos de la Lista del Producto</b>	
<b>Autor:</b> Equipo de Desarrollo (M.R. y A.G.)	
<b>Fecha:</b> 29 Julio 2016	
<b>Estimación restante:</b> 4 días	
<b>R13:</b> Crear aulas virtuales	
<b>Puntos de historias:</b> 18	
<b>Estimación total:</b> 6 días	
<b>Estimación restante:</b> 6 días	
<b>R14:</b> Revisar el SI como otro usuario	
<b>Puntos de historias:</b> 6	
<b>Estimación total:</b> 2 días	
<b>Estimación restante:</b> 2 días	

**Producto de Verificación 14:** Informe de la estimación de los elementos de la Lista del Producto.

#### 4.4.6.4. Tarea 2.2.4. Priorizar requisitos

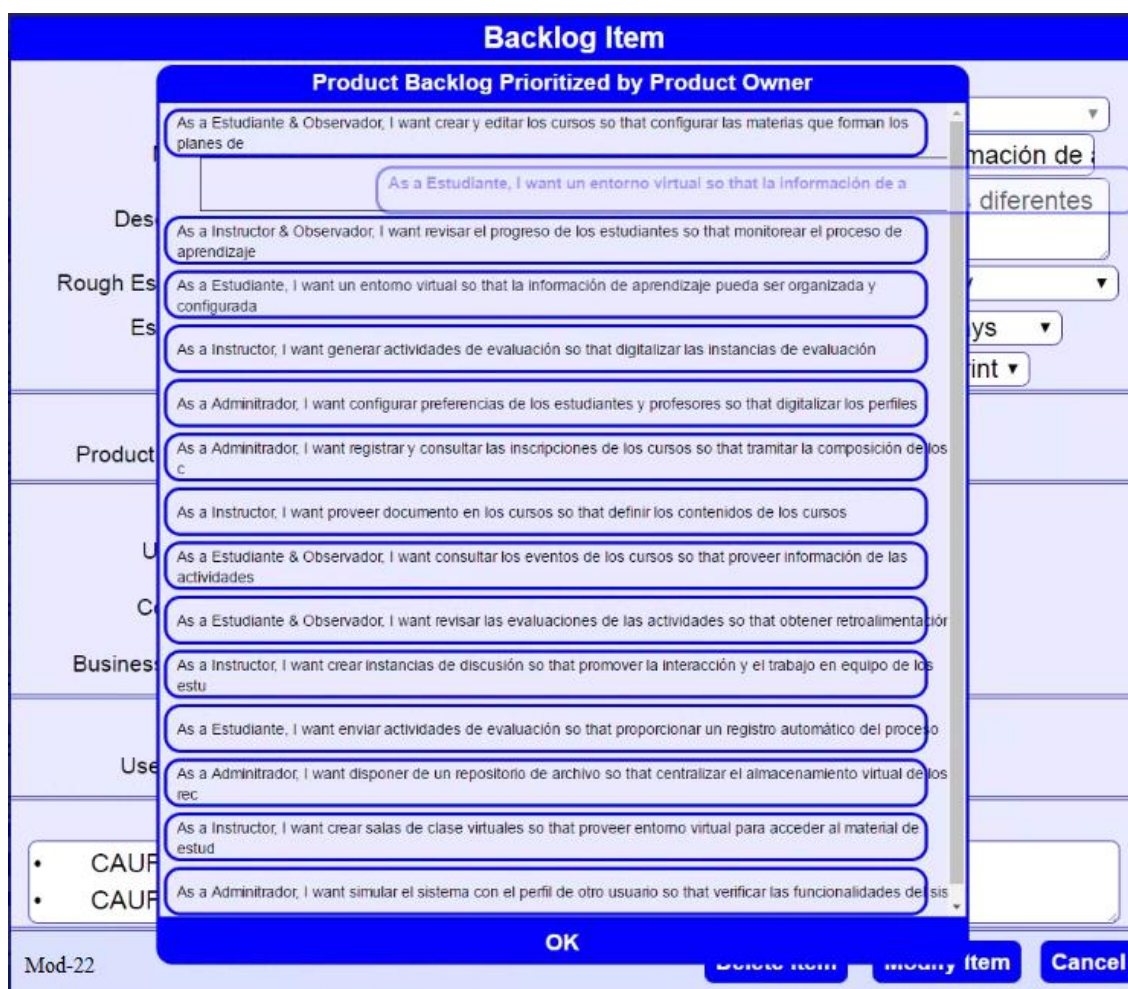
**Descripción:** El objetivo de esta tarea es priorizar los requisitos, en base a los elementos de los aspectos relevantes del SI de gestión de aprendizaje, es decir, en esta tarea se priorizan los requisitos mediante el enfoque del método de priorización QMPSR, presentado en la sección 3.3. Asimismo, los requisitos también identifican la prioridad que tienen para el Dueño del Producto y el Arquitecto de la Información de manera independiente.

En la elaboración de la Tarea 2.2.4, también se manipulará el módulo “Gestionar Historias de usuarios (Mod-22)” (ver Figura 58). Este módulo permite asociar los aspectos relevantes de cada requisito (identificados y definidos en la Tarea 1.4.1), e identificar la prioridad del Dueño del Producto y el Arquitecto de la Información respecto a los requisitos de manera independiente.

Así pues, la priorización de los requisitos en base a los elementos de los aspectos relevantes (es decir, utilizando el enfoque de QMPSR) se lleva a cabo a través del apartado *Related Aspects* del módulo “Gestionar Historias de usuarios (Mod-22)” (ver Figura 58). En este apartado del módulo, se listan los diferentes elementos y aspectos identificados en el Informe final sobre los elementos y los aspectos relevantes del proyecto (P8) de la Tarea 1.4.1. En la Figura 58, se muestra la priorización del requisito Gestionar cursos en un entorno integrado (R2) a través del módulo “Gestionar Historias de usuarios (Mod-22)”, donde se han asignado los siguientes elementos: “Proveer adaptabilidad del entorno de aprendizaje” del aspecto relevante Usabilidad, “Descripción de las entidades de conocimiento” del

aspecto relevante Contenido y “Gestionar el progreso de los estudiantes” del aspecto relevante Valor de Negocio.

Por otro lado, la priorización de los requisitos, entregada por el Dueño del Producto y el Arquitecto de la Información de manera separada, se lleva a cabo mediante el apartado Priority del módulo “Gestionar Historias de usuarios (Mod-22)” (ver Figura 58). Este apartado del módulo permite al Dueño del Producto y al Arquitecto de la Información priorizar los requisitos de manera independiente, en base a sus preferencias. La Figura 59 muestra la configuración de la priorización de los requisitos por parte del Dueño del Producto a través del módulo “Gestionar Historias de usuarios (Mod-22)”, que se accede mediante el botón Set Up. En este caso, los requisitos se priorizan en términos relativos al conjunto total de requisitos, es decir, la posición del requisito en el listado representa su orden de prioridad (siendo el requisito con mayor prioridad el que se encuentra en el tope de la lista).



**Figura 59:** Configuración de la priorización de los requisitos por parte del Dueño del Producto a través del módulo “Gestionar Historias de usuarios (Mod-22)”.

**Información de entrada:** Las fuentes principales de información corresponden al Catálogo de requisitos analizados del proyecto (P11) de la Tarea 2.1.2, a la Lista del Producto con los elementos descritos en formato de Historia de Usuario (P13), proporcionada por la Tarea 2.2.2, y al Informe final sobre los elementos y los aspectos relevantes del proyecto.

**Producto de salida:** El principal resultado de la Tarea 2.2.4 corresponde a un Informe de las prioridades de los elementos de la Lista del Producto (P15). Del mismo modo que la Tarea 2.2.2 y la Tarea 2.2.3, los datos de este producto también se pueden generar y consultar en el módulo “Gestionar Historias de usuarios (Mod-22)” (ver Figura 58). Los detalles de este producto (P15) se presentan a continuación:

<b>P15. Informe de las prioridades de los elementos de la Lista del Producto</b>		
<b>Autor:</b> Dueño del Producto (J.P.)		
<b>Fecha:</b> 29 Julio 2016		
<b>Identificador</b>	<b>Aspectos relevantes</b>	<b>Prioridad</b>
12	Usabilidad	Alta
13	Contenido	Media
14	Valor de Negocio	Baja
<b>R1:</b> Crear y editar cursos <b>Prioridad Arquitecto de la Información:</b> 50 <b>Prioridad Dueño del Producto:</b> 45 <b>Elementos de los aspectos asociados:</b> <ul style="list-style-type: none"> <li>• <b>AE12.5:</b> Contemplar los potenciales errores</li> <li>• <b>AE13.5:</b> Preferencias de los estudiantes</li> <li>• <b>AE13.6:</b> Descripción de las entidades de conocimiento</li> <li>• <b>AE14.4:</b> Gestionar las entidades de conocimiento</li> <li>• <b>AE14.6:</b> Coordinar composición de las entidades de conocimiento</li> </ul> <b>Clasificación QMPSR:</b> 10,83		
<b>R2:</b> Gestionar cursos en un entorno integrado <b>Prioridad Arquitecto de la Información:</b> 55 <b>Prioridad Dueño del Producto:</b> 46 <b>Elementos de los aspectos asociados:</b> <ul style="list-style-type: none"> <li>• <b>AE12.1:</b> Proveer adaptabilidad al entorno de aprendizaje</li> </ul>		



**P15. Informe de las prioridades de los elementos de la Lista del Producto**

**Autor:** Dueño del Producto (J.P.)

**Fecha:** 29 Julio 2016

- **AE12.2:** Proveer facilidad de aprendizaje
- **AE12.6:** Proveer facilidad del SI para que éste sea recordado
- **AE13.6:** Descripción de las entidades de conocimiento
- **AE14.1:** Gestionar el progreso de los estudiantes

**Clasificación QMPSR:** 12,17

**R3:** Crear y editar actividades de evaluación

**Prioridad Arquitecto de la Información:** 45

**Prioridad Dueño del Producto:** 47

**Elementos de los aspectos asociados:**

- **AE12.5:** Contemplar los potenciales errores
- **AE13.7:** Retroalimentación de las actividades de evaluación
- **AE14.3:** Monitorear el proceso de aprendizaje

**Clasificación QMPSR:** 9,67

**R4:** Configurar perfiles de estudiantes y profesores

**Prioridad Arquitecto de la Información:** 30

**Prioridad Dueño del Producto:** 20

**Elementos de los aspectos asociados:**

- **AE12.3:** Proveer accesibilidad a la información (opciones de formato)
- **AE13.1:** Datos personales de los estudiantes
- **AE13.3:** Datos personales de los instructores

**Clasificación QMPSR:** 4,33

**R5:** Gestionar las inscripciones en los cursos

**Prioridad Arquitecto de la Información:** 33

**Prioridad Dueño del Producto:** 55

**Elementos de los aspectos asociados:**

- **AE14.2:** Gestionar la matrícula

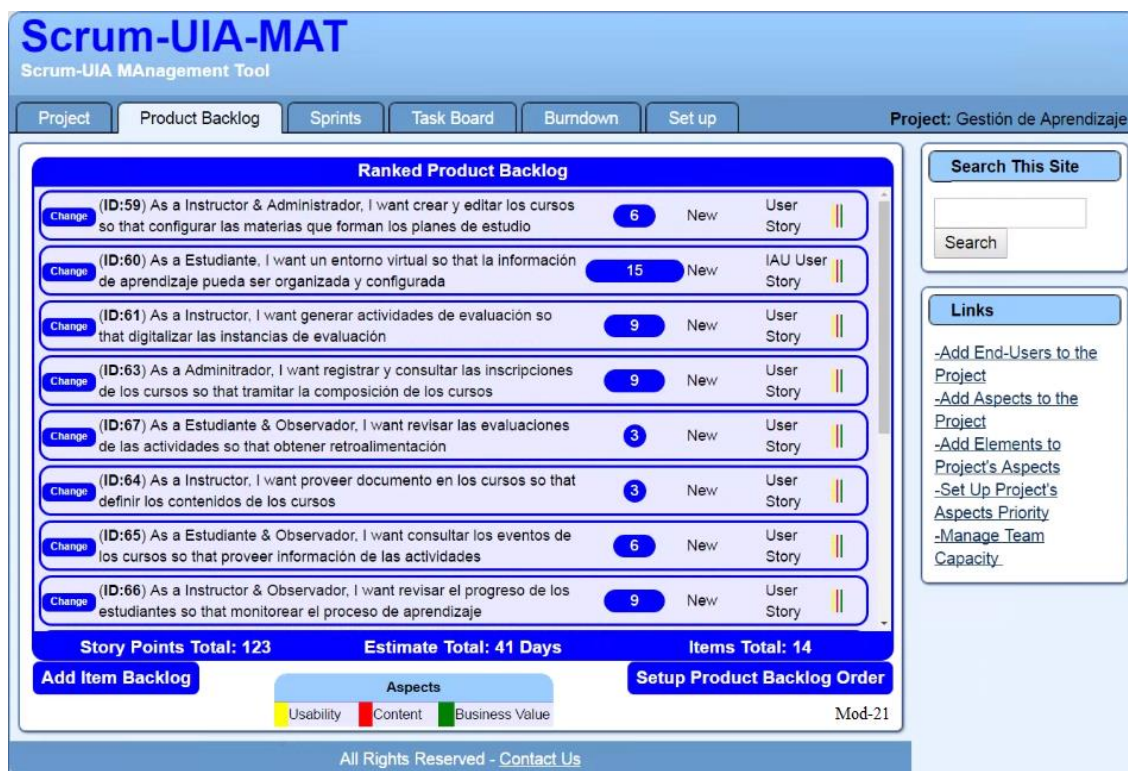
<p><b>P15. Informe de las prioridades de los elementos de la Lista del Producto</b></p> <p><b>Autor:</b> Dueño del Producto (J.P.)</p> <p><b>Fecha:</b> 29 Julio 2016</p> <ul style="list-style-type: none"> <li>• <b>AE14.5:</b> Organizar el almacenamiento de la información</li> <li>• <b>AE14.6:</b> Coordinar composición de las entidades de conocimiento</li> </ul> <p><b>Clasificación QMPSR:</b> 3,17</p>
<p><b>R6:</b> Incorporar material en los cursos</p> <p><b>Prioridad Arquitecto de la Información:</b> 37</p> <p><b>Prioridad Dueño del Producto:</b> 53</p> <p><b>Elementos de los aspectos asociados:</b></p> <ul style="list-style-type: none"> <li>• <b>AE12.5:</b> Contemplar los potenciales errores</li> <li>• <b>AE13.2:</b> Contenido de las entidades de conocimiento</li> <li>• <b>AE14.6:</b> Coordinar composición de las entidades de conocimiento</li> </ul> <p><b>Clasificación QMPSR:</b> 9,17</p>
<p><b>R7:</b> Consultar los eventos y tareas de los cursos</p> <p><b>Prioridad Arquitecto de la Información:</b> 39</p> <p><b>Prioridad Dueño del Producto:</b> 51</p> <p><b>Elementos de los aspectos asociados:</b></p> <ul style="list-style-type: none"> <li>• <b>AE12.3:</b> Proveer accesibilidad a la información (opciones de formato)</li> <li>• <b>AE12.4:</b> Reducir la carga mental de trabajo</li> </ul> <p><b>Clasificación QMPSR:</b> 7</p>
<p><b>R8:</b> Consultar información sobre el progreso de los estudiantes</p> <p><b>Prioridad Arquitecto de la Información:</b> 29</p> <p><b>Prioridad Dueño del Producto:</b> 52</p> <p><b>Elementos de los aspectos asociados:</b></p> <ul style="list-style-type: none"> <li>• <b>AE12.3:</b> Proveer accesibilidad a la información (opciones de formato)</li> <li>• <b>AE13.8:</b> Registro de las actividades de los estudiantes</li> </ul> <p><b>Clasificación QMPSR:</b> 7,83</p>
<p><b>R9:</b> Consultar evaluaciones de las actividades</p>

<b>P15. Informe de las prioridades de los elementos de la Lista del Producto</b> <b>Autor:</b> Dueño del Producto (J.P.) <b>Fecha:</b> 29 Julio 2016
<b>Prioridad Arquitecto de la Información:</b> 41 <b>Prioridad Dueño del Producto:</b> 46 <b>Elementos de los aspectos asociados:</b> <ul style="list-style-type: none"><li>• <b>AE13.6:</b> Descripción de las entidades de conocimiento</li><li>• <b>AE13.7:</b> Retroalimentación de las actividades de evaluación</li><li>• <b>AE14.3:</b> Monitorear el proceso de aprendizaje</li><li>• <b>AE14.6:</b> Coordinar composición de las entidades de conocimiento</li></ul> <b>Clasificación QMPSR:</b> 7,33
<b>R10:</b> Enviar, de manera electrónica, las actividades de evaluación <b>Prioridad Arquitecto de la Información:</b> 33 <b>Prioridad Dueño del Producto:</b> 27 <b>Elementos de los aspectos asociados:</b> <ul style="list-style-type: none"><li>• <b>AE12.5:</b> Contemplar los potenciales errores</li><li>• <b>AE13.8:</b> Registro de las actividades de los estudiantes</li></ul> <b>Clasificación QMPSR:</b> 7,67
<b>R11:</b> Crear entornos de discusión <b>Prioridad Arquitecto de la Información:</b> 29 <b>Prioridad Dueño del Producto:</b> 28 <b>Elementos de los aspectos asociados:</b> <ul style="list-style-type: none"><li>• <b>AE13.4:</b> Comunicaciones entre estudiantes e instructores</li></ul> <b>Clasificación QMPSR:</b> 3
<b>R12:</b> Proveer un repositorio de archivos <b>Prioridad Arquitecto de la Información:</b> 10 <b>Prioridad Dueño del Producto:</b> 17 <b>Elementos de los aspectos asociados:</b> <ul style="list-style-type: none"><li>• <b>AE14.5:</b> Organizar el almacenamiento de la información</li></ul>

<b>P15. Informe de las prioridades de los elementos de la Lista del Producto</b> <b>Autor:</b> Dueño del Producto (J.P.) <b>Fecha:</b> 29 Julio 2016
<ul style="list-style-type: none"> <li>• <b>AE14.6:</b> Coordinar composición de las entidades de conocimiento</li> </ul> <b>Clasificación QMPSR:</b> 2,67
<b>R13:</b> Crear aulas virtuales <b>Prioridad Arquitecto de la Información:</b> 7 <b>Prioridad Dueño del Producto:</b> 18 <b>Elementos de los aspectos asociados:</b> <ul style="list-style-type: none"> <li>• <b>AE12.3:</b> Proveer accesibilidad a la información (opciones de formato)</li> <li>• <b>AE14.1:</b> Gestionar el progreso de los estudiantes</li> </ul> <b>Clasificación QMPSR:</b> 7,50
<b>R14:</b> Revisar el SI como otro usuario <b>Prioridad Arquitecto de la Información:</b> 5 <b>Prioridad Dueño del Producto:</b> 7 <b>Elementos de los aspectos asociados:</b> <ul style="list-style-type: none"> <li>• <b>AE12.3:</b> Proveer accesibilidad a la información (opciones de formato)</li> </ul> <b>Clasificación QMPSR:</b> 6,50

**Producto de Verificación 15:** Informe de las prioridades de los elementos de la Lista del Producto.

Finalmente, el módulo “Consultar Historias de usuarios (Mod-21)” de Scrum-UIA-MAT permite revisar los requisitos del SI de gestión de aprendizaje, ordenados de acuerdo a las prioridades entregadas en la Tarea 2.2.4. La Figura 60 muestra la Lista del Producto con los requisitos descritos (Tarea 2.2.2), estimados (Tarea 2.2.3) y priorizados (Tarea 2.2.4), utilizando el módulo “Consultar Historias de usuarios (Mod-21)”. Cabe destacar que los requisitos, los elementos y los aspectos definidos en esta verificación son los mismos que se emplearon en el ejemplo de aplicación de QMPSR (ver sección 3.3.2), generando en ambas situaciones (ejemplo de aplicación y de verificación) las mismas relaciones entre los requisitos y los elementos de los aspectos. Por lo tanto, los resultados del ejemplo de aplicación de QMPSR permiten observar en detalle la capacidad de este método para conducir el proceso de priorización de Scrum-UIA, de acuerdo a las preferencias del proyecto, sin generar colisiones de requisitos.



**Figura 60:** Consulta de la Lista del Producto con los requisitos descritos, estimados y priorizados del SI de gestión de aprendizaje, mediante el uso del módulo “Consultar Historias de usuarios (Mod-21)”.

#### 4.4.7. Actividad 2.3. Determinación de la definición de “Terminado”

Una vez configurada la Lista del Producto del SI de gestión de aprendizaje (Actividad 2.2), en esta actividad se identificará y determinará una definición de “Terminado” para las tareas que el Equipo de Desarrollo llevará a cabo en el Sprint. El objetivo de obtener una definición de “Terminado” es facilitar el seguimiento de las tareas de desarrollo y conocer el momento apropiado para dar por finalizadas las tareas.

##### 4.4.7.1. Tarea 2.3.1. Acordar una definición de “Terminado”

**Descripción:** En esta tarea se acuerda una definición de “Terminado” para las tareas que se llevan a cabo en el siguiente Sprint.

En la elaboración de la Tarea 2.3.1, se utilizará el módulo “Agregar Sprint (Mod-32)”. Este módulo permite registrar la definición de “Terminado” para cada Sprint y describir sus datos generales. En la Figura 61, se muestra la determinación de la definición de “Terminado” (ver campo Definition of Done) para el primer Sprint del SI de gestión de aprendizaje.

**Sprint**

Project: Learning Management

Sprint Name: Sprint 1

Description: Sprint initial project

Objective: Develop the elements related to the end-users' priorities that allow to drive incremental development through the deliverables of information architecture.

Definition of Done: Requirements meet the end-user acceptance criteria

Start Date: 

June 2016						
Mo	Tu	We	Th	Fri	Sat	Sun
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

End Date: 

June 2016						
Mo	Tu	We	Th	Fri	Sat	Sun
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

Mod-32 Add Sprint Cancel

**Figura 61:** Configuración del Sprint mediante el módulo “Agregar Sprint (Mod-32)”.

**Información de entrada:** La información de entrada de esta tarea es la Lista del Producto con los elementos descritos en formato de Historia de Usuario (P13) de la Tarea 2.2.2, y la priorización de los elementos de la Lista del Producto (P15), proporcionada por la Tarea 2.2.4.

### P16. Informe sobre la definición de “Terminado” para el Sprint

**Autor:** Dueño del Producto (J.P.)

**Fecha:** 31 Julio 2016

#### S1: Sprint 1

**Descripción del Sprint:** Sprint inicial del SI de gestión de aprendizaje

#### Definición de “Terminado”:

- Las historias de usuarios deben cumplir con los criterios de aceptación de los usuarios finales
- El diseño de las pantallas debe cumplir con la estructura de las maquetas (Wireframes)
- Los datos proporcionados deben considerar alternativas de accesibilidad

**Producto de Verificación 16:** Informe sobre la definición de “Terminado” para el Sprint.

**Producto de salida:** A continuación se presenta el Informe sobre la definición de “Terminado” para el Sprint (P16) que se generó en la Tarea 2.3.1. Este producto utiliza como fuente principal los productos P13 y P15. Cabe destacar que este producto también se puede consultar a través del módulo “Agregar Sprint (Mod-32)” (ver Figura 61).

#### 4.4.8. Actividad 2.4. Determinación del objetivo del Sprint

Una vez acordada una definición de “Terminado” para el Sprint (Actividad 2.3), en esta actividad se identificará y determinará el objetivo inicial que se propone para el siguiente Sprint, con el fin de dirigir sus prioridades.

##### 4.4.8.1. Tarea 2.4.1. Establecer un objetivo para el Sprint

**Descripción:** El propósito de esta tarea es acordar un objetivo del Sprint.

Al igual que en la Tarea 2.3.1, en la ejecución de la Tarea 2.4.1 también se utilizará el módulo “Agregar Sprint (Mod-32)”. Este módulo permite registrar el objetivo inicial de cada Sprint del proyecto. En la Figura 61, se puede observar la determinación del objetivo (ver campo *Objective*) para el primer Sprint del SI de gestión de aprendizaje.

**Información de entrada:** Esta tarea utiliza como fuente de información la Lista del Producto con los elementos descritos en formato de Historia de Usuario (P13) de la Tarea 2.2.2, y la estimación de los elementos de la Lista del Producto (P14) de la Tarea 2.2.4.

**Producto de salida:** A continuación se presenta el Informe sobre el objetivo inicial para el Sprint (P17) que se generó en la Tarea 2.4.1. Este producto utiliza como fuente principal los productos P13 y P14. Al igual que la Tarea 2.3.1, este producto también se puede examinar con el módulo “Agregar Sprint (Mod-32)” (ver Figura 61).

P17. Informe sobre el objetivo inicial para el Sprint	
<b>Autor:</b> Dueño del Producto (J.P.)	
<b>Fecha:</b> 31 Julio 2016	
S1: Sprint 1	
<b>Objetivo del Sprint:</b> Proveer las bases arquitectónicas (diseño estructural del contenido y modelo de navegación) del SI de gestión de aprendizaje.	

**Producto de Verificación 17:** Informe sobre el objetivo inicial para el Sprint.

#### 4.4.9. Actividad 3.1. Determinación de la capacidad del Equipo de Desarrollo

Una vez identificados los requisitos (Actividad 2.2), la definición de “Terminado” (Actividad 2.3) y el objetivo inicial del Sprint (Actividad 2.4), en esta actividad se procederá a identificar la disponibilidad de recursos y la capacidad del Equipo de Desarrollo.

#### 4.4.9.1. Tarea 3.1.1. Determinar la duración del Sprint

**Descripción:** El objetivo de esta tarea es identificar la duración del siguiente Sprint.

Del mismo modo que en las tareas 2.3.1 y 2.4.1, en esta tarea también se utilizará el módulo “Agregar Sprint (Mod-32)”. Este módulo permite registrar la duración de cada Sprint del proyecto. En la Figura 61, se puede observar la duración indicada para el primer Sprint del SI de gestión de aprendizaje (ver campos `Start Date` y `End Date`).

**Información de entrada:** Esta tarea utiliza como fuente de información externa la duración de los Sprints anteriores (E6).

**Producto de salida:** A continuación se presenta el Informe sobre la duración del Sprint (P18) que se generó en la Tarea 3.1.1. Al igual que en la Tarea 2.3.1 y la Tarea 2.4.1, este producto también se puede revisar a través del módulo “Agregar Sprint (Mod-32)” (ver Figura 61).

P18. Informe sobre la duración del Sprint	
<b>Autor:</b> Equipo de Desarrollo (M.R. y A.G.)	
<b>Fecha:</b> 31 Julio 2016	
S1: Sprint 1	
<b>Inicio del Sprint:</b> 8 de Agosto del 2017	
<b>Fin del Sprint:</b> 19 de Agosto del 2017	

**Producto de Verificación 18:** Informe sobre la duración del Sprint.

#### 4.4.9.2. Tarea 3.1.2. Determinar la disponibilidad de recursos para el Equipo de Desarrollo

**Descripción:** Una vez identificada la duración del Sprint, esta tarea determinará la disponibilidad de los recursos humanos del Equipo de Desarrollo para el siguiente Sprint.

En la realización de la Tarea 3.1.2, se utilizará el módulo “Gestionar la Capacidad del Equipo (Mod-66)” (ver Figura 62). Este módulo permite registrar y consultar la disponibilidad de los recursos humanos del Equipo de Desarrollo. En la Figura 62, se puede observar la determinación de los días-persona disponibles de los miembros del Equipo de Desarrollo para el siguiente y anterior Sprint (ver apartado `Available Person-Days` del módulo).



**Scrum-UIA-MAT**  
Scrum-UIA Management Tool

Project | Product Backlog | Sprints | Task Board | Burndown | Set up | Project: Gestión de Aprendizaje

### Manage Team

**Available Person-Days**

Name	Next Sprint	Previous Sprint	Type
<a href="#">Edit</a> Adams Galkus	5	4	Development Team
<a href="#">Edit</a> Luis Rojas	6	7	Information Architect
<a href="#">Edit</a> Mario Reyes	4	6	Development Team
<b>Total</b>	<b>15</b>	<b>17</b>	

**Real Velocity:**

**Factor Dedication:** (Real Velocity / Available Previous Sprint)

**Estimated Velocity:** (Factor Dedication \* Available Next Sprint)

**Add Team Member**

Name	<input type="text"/>
Next Sprint	<input type="text"/>
Previous Sprint	<input type="text"/>
Type	<input type="text"/>

[Insert](#) [Cancel](#)

Mod-66

**Search This Site**

[Search](#)

**Links**

- Add End-Users to the Project
- Add Aspects to the Project
- Add Elements to Project's Aspects
- Set Up Project's Aspects Priority
- Manage Team Capacity

**Figura 62:** Determinación de la capacidad del Equipo de Desarrollo, mediante el módulo “Gestionar la Capacidad del Equipo (Mod-66)”.

**Información de entrada:** Esta tarea utiliza como fuentes de entrada el Informe sobre la duración del Sprint (P18) de la Tarea 3.1.1 y las capacidades disponibles del Equipo de Desarrollo (E7) que permiten calcular y determinar las disponibilidades de cada miembro del Equipo de Desarrollo para el siguiente Sprint.

**Producto de salida:** El principal resultado de esta tarea es el Informe sobre la disponibilidad del Equipo de Desarrollo para el Sprint (P19). Este producto también se puede consultar mediante el módulo “Gestionar la Capacidad del Equipo (Mod-66)” (ver Figura 62) y se presenta en detalle a continuación:

<b>P19. Informe sobre la disponibilidad del Equipo de Desarrollo para el Sprint</b>	
<b>Autor:</b> Equipo de Desarrollo (M.R. y A.G.)	
<b>Fecha:</b> 31 Julio 2016	
<b>S1: Sprint 1</b>	
<b>MED1:</b> Adams Galkus	
<b>Disponibilidad para el siguiente Sprint:</b> 5	
<b>Disponibilidad en el Sprint anterior:</b> 4	
<b>MED2:</b> Luis Rojas	

<b>Disponibilidad para el siguiente Sprint: 6</b>
<b>Disponibilidad en el Sprint anterior: 7</b>
<b>MED3: Mario Reyes</b>
<b>Disponibilidad para el siguiente Sprint: 4</b>
<b>Disponibilidad en el Sprint anterior: 6</b>
<b>Disponibilidad Total para el siguiente Sprint: 15</b>
<b>Disponibilidad Total en el Sprint anterior: 17</b>

**Producto de Verificación 19:** Informe sobre la disponibilidad del Equipo de Desarrollo para el Sprint.

#### 4.4.9.3. Tarea 3.1.3. Determinar la velocidad del Equipo de Desarrollo

**Descripción:** Una vez identificada la disponibilidad de los recursos humanos del Equipo de Desarrollo, en esta tarea se determinará su velocidad para el siguiente Sprint.

En la elaboración de la Tarea 3.1.3, también se manipulará el módulo “Gestionar la Capacidad del Equipo (Mod-66)”. Este módulo permite ingresar y calcular la capacidad del Equipo de Desarrollo para el siguiente Sprint. Como se puede observar en la Figura 62, el Equipo de Desarrollo tiene una velocidad real de 50 puntos de historias, un factor de dedicación de 2,9 y una velocidad estimada de 43,5 puntos de historias (ver campos *Real Velocity*, *Factor Dedication* y *Estimated Velocity*, respectivamente).

**Información de entrada:** La Tarea 3.1.3 utiliza como fuentes de entrada la velocidad histórica del Equipo de Desarrollo (E8) de los Sprints anteriores, el Informe sobre la duración del Sprint (P18) de la Tarea 3.1.1, y el Informe sobre la disponibilidad del Equipo de Desarrollo para el Sprint (P19) de la Tarea 3.1.2.

**Producto de salida:** La Tarea 3.1.3 genera como resultado el Informe sobre la velocidad del Equipo de Desarrollo (P20). Al igual que en la Tarea 3.1.2, este producto también se puede consultar mediante el módulo “Gestionar la Capacidad del Equipo (Mod-66)” (ver Figura 62) y se presenta en detalle a continuación:

<b>P20. Informe sobre la velocidad del Equipo de Desarrollo</b>	
<b>Autor:</b> Equipo de Desarrollo (M.R. y A.G.)	
<b>Fecha:</b> 31 Julio 2016	
<b>S1: Sprint 1</b>	
<b>Velocidad real:</b> 50	
<b>Factor de dedicación:</b> 2,9 [ <i>Velocidad Real / Total Disponibilidad del Sprint anterior (P19)</i> ]	
<b>Velocidad estimada:</b> 43,5 [ <i>Factor de dedicación * Total Disponibilidad para el siguiente Sprint (P19)</i> ]	

**Producto de Verificación 20:** Informe sobre la velocidad del Equipo de Desarrollo.

#### 4.4.10. Actividad 3.2. Refinamiento del objetivo del Sprint

Una vez identificada la capacidad del Equipo de Desarrollo para el siguiente Sprint (Actividad 3.1), en esta actividad se determinará el objetivo que definitivamente se considerará en el siguiente Sprint.

##### 4.4.10.1. Tarea 3.2.1. Refinar el objetivo para el Sprint

**Descripción:** La finalidad de esta tarea es refinar el objetivo del Sprint inicialmente propuesto.

La Tarea 2.4.1 se realizará a través del módulo “Agregar Sprint (Mod-32)” (ver Figura 61) que permite modificar el objetivo inicial del Sprint.

**Información de entrada:** Esta tarea utiliza como fuente de información el objetivo inicialmente propuesto (P17) en la Tarea 2.4.1, la Lista del Producto con los elementos descritos en formato de Historia de Usuario (P13) de la Tarea 2.2.2, y la estimación de los elementos de la Lista del Producto (P14) de la Tarea 2.2.4.

**Producto de salida:** El principal resultado de esta tarea es el Informe sobre el objetivo refinado del Sprint (P21) que utiliza como fuente principal los productos P13, P14 y P17. El objetivo inicial del Sprint, propuesto en la Tarea 2.4.1, se modificará para incorporar temas relacionados con la conducción del proceso de desarrollo. Al igual que en la Tarea 2.3.1 y la Tarea 2.4.1, este producto también se puede examinar con el módulo “Agregar Sprint (Mod-32)” (ver Figura 61) y se presenta en detalle a continuación:

<b>P21. Informe sobre el objetivo refinado del Sprint</b>	
<b>Autor:</b> Dueño del Producto (J.P.)	
<b>Fecha:</b> 31 Julio 2016	
<b>S1:</b> Sprint 1	
<b>Objetivo del Sprint:</b> Proveer las bases arquitectónicas (diseño estructural del contenido y modelo de navegación) del SI de gestión de aprendizaje, para conducir el desarrollo de los siguientes Sprints.	

**Producto de Verificación 21:** Informe sobre el objetivo refinado del Sprint.

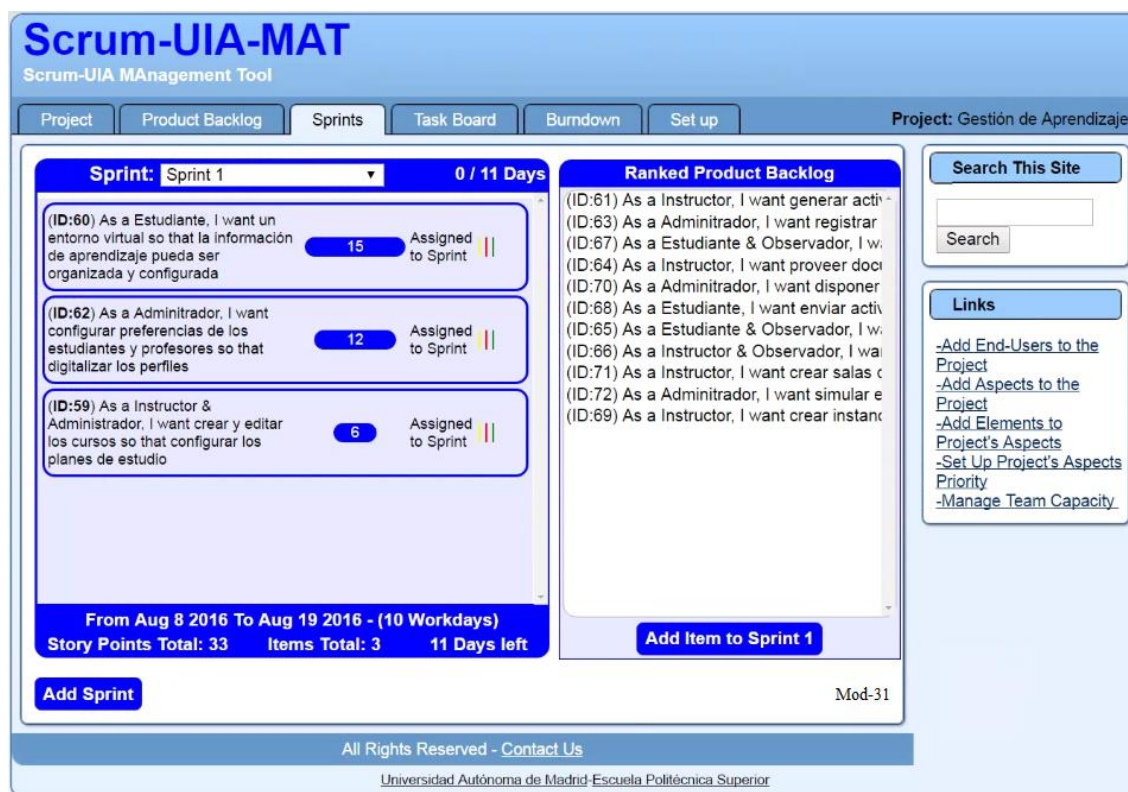
#### 4.4.11. Actividad 3.3. Selección de requisitos para el Sprint

Una vez determinada la capacidad del Equipo de Desarrollo (Actividad 3.1) y definido el objetivo del Sprint (Actividad 3.2), en esta tarea se seleccionarán los requisitos del SI de gestión de aprendizaje que se implementarán en el primer Sprint.

##### 4.4.11.1. Tarea 3.3.1. Escoger requisitos para el Sprint

**Descripción:** El objetivo de esta tarea es seleccionar los requisitos de la Lista del Producto que formarán la Lista de Pendientes del Sprint. Para ello, se tomará en consideración la capacidad del Equipo de Desarrollo para el siguiente Sprint (Actividad 3.1), el objetivo del Sprint (Actividad 3.2) y las prioridades de los requisitos (Actividad 2.2).

En la realización de la tarea 3.3.1, se utilizará el módulo “Configurar Sprints (Mod-31)” de Scrum-UIA-MAT. Este módulo permite establecer y configurar los requisitos que se implementarán en el siguiente Sprint (Lista de Pendientes del Sprint). En la Figura 63, se presenta la configuración de la Lista de Pendientes del Sprint del SI de gestión de aprendizaje para el primer Sprint. Como se puede observar, en el apartado *Ranked Product Backlog*, se listan todos los requisitos, ordenados de acuerdo a la priorización entregada en la Tarea 2.2.4. Por lo tanto, se seleccionarán los requisitos para formar la Lista de Pendientes del Sprint, hasta alcanzar la velocidad estimada del Equipo de Desarrollo (43,5 puntos de historias).



**Figura 63:** Configuración de la Lista de Pendientes del Sprint mediante el módulo “Configurar Sprints (Mod-31)”.

P22. Lista de Pendientes del Sprint con los requisitos para el Sprint	
<b>Autor:</b> Equipo de Desarrollo (M.R. y A.G.)	
<b>Fecha:</b> 31 Julio 2016	
S1: Sprint 1	
<b>Requisitos:</b> <ul style="list-style-type: none"> <li>• <b>R1:</b> Crear y editar cursos</li> <li>• <b>R2:</b> Gestionar cursos en un entorno integrado</li> <li>• <b>R4:</b> Configurar perfiles de estudiantes y profesores</li> </ul>	
<b>Total de Puntos de Historias:</b> 33 <b>Total de Días-Persona:</b> 11 <b>Total de Requisitos:</b> 3	

**Producto de Verificación 22:** Lista de Pendientes del Sprint con los requisitos para el Sprint.

**Información de entrada:** En la Tarea 3.3.1, el Equipo de Desarrollo trabaja sobre una Lista del Producto que se encuentra descrita en formato de historias de usuarios, estimada y priorizada (los productos P13, P14 y P15 de las tareas 2.2.2, 2.2.3 y 2.2.4,

respectivamente). Asimismo, los requisitos son escogidos de acuerdo a las prioridades establecidas en la Tarea 2.2.4, considerando el objetivo del Sprint (P21 de la Tarea 3.2.1) y hasta alcanzar la capacidad del Equipo de Desarrollo (P20 de la Tarea 3.1.3).

**Producto de salida:** El principal resultado de la Tarea 3.3.1 es una Lista de Pendientes del Sprint con los requisitos para el Sprint (P22). Este producto también se puede consultar con el módulo “Configurar Sprints (Mod-31)” (ver Figura 63) y se presenta en detalle a continuación:

#### **4.4.12.Actividad 3.4. Generación de tareas de desarrollo**

Una vez identificados los requisitos para la Lista de Pendientes del Sprint (Actividad 3.3), se configurará (se identificarán, describirán y estimarán) las tareas requeridas para implementar los requisitos durante la Ejecución del Sprint.

##### **4.4.12.1. Tarea 3.4.1. Establecer tareas para los requisitos seleccionados**

**Descripción:** El objetivo de esta tarea es determinar las tareas requeridas para implementar los requisitos seleccionados para el siguiente Sprint (Tarea 3.3.1), e identificar las tareas de desarrollo relacionadas con la AI que permiten conducir el desarrollo incremental, dirigido por los niveles de fidelidad de los entregables de la AI.

En la ejecución de la Tarea 3.4.1, se utilizará el módulo “Gestionar Tareas (Mod-42)” de Scrum-UIA-MAT. Este módulo permite registrar y consultar las tareas de desarrollo identificadas para cada requisito. En la Figura 64, se muestra la configuración de la tarea de desarrollo Modelo de contenido de estudiantes e instructores (ver campo Task) para el requisito “R4: Configurar perfiles de estudiantes y profesores” (ver campo Backlog Item).

**Información de entrada:** La Tarea 3.4.1 utiliza como fuentes de información el Modelo de contenido general del proyecto (P4) de la Tarea 1.2.1, la Lista de Pendientes del Sprint con los requisitos para el Sprint (P22) de la Tarea 3.3.1, y el Informe sobre la definición de “Terminado” para el Sprint (P16) de la Tarea 2.3.1. Estos productos permiten determinar las tareas de desarrollo, considerando además su alcance, a través de la definición de “Terminado” acordada anteriormente. Cabe destacar que la utilización de la herramienta InterArch, en la elaboración del Modelo de contenido general del proyecto (P4) durante la Tarea 1.2.1, permite generar automáticamente información de diseño (el diagrama de clases UML, P24) que se utiliza en esta fase de Scrum-UIA para identificar tareas de desarrollo y, además, apoyar su elaboración durante la Ejecución del Sprint.

**Producto de salida:** La Tarea 3.4.1 tiene, como salida principal, el Informe de las tareas de desarrollo de los requisitos del Sprint (P23) y el Diagrama de Clases UML (P24). El producto P23 también se puede consultar con el módulo “Gestionar Tareas (Mod-42)” (ver Figura 64). A continuación, se describen en detalle estos productos:

**Manage Tasks**

Backlog Item: (ID:62) As a Adminitrador, I want configurar preferencias de los estudiantes ! ▾

Task: Modelo de contenido de estudiantes e instructores ▾

Name: Modelo de contenido de estudiantes e instructores

Description:

Estimate: 6 Hours ▾ Assigned to Luis Rojas ▾

Status: To do ▾

**Incorporate Information Architecture Agile Techniques**

IA Techniques

**Filter Techniques**

Affinity diagram  
 Background investigation  
 Benchmarking  
 Card Sorting

Add
Remove

Added IA Techniques

InterArch-T

Mod-42

Accept
Cancel

**Figura 64:** Configuración de las tareas de desarrollo, utilizando el módulo “Gestionar Tareas (Mod-42)”.

P23. Informe de las tareas de desarrollo de los requisitos del Sprint	
<b>Autor:</b> Equipo de Desarrollo (M.R. y A.G.)	
<b>Fecha:</b> 01 Agosto 2016	
S1: Sprint 1	
<b>Requisito:</b> R1: Crear y editar cursos	
<b>Tareas de Desarrollo:</b>	
TD1. 1: Maqueta (wireframe) para crear cursos	
TD1. 2: Diagrama de clases de cursos	
TD1. 3: Programación de algoritmo para crear cursos	
TD1. 4: Algoritmo para registrar actividad de los estudiantes	

**Requisito: R2:** Gestionar cursos en un entorno integrado

**Tareas de Desarrollo:**

TD2. 1: Inventario de contenido (Estudiante, Curso, Instructores)

TD2. 2: Vocabulario controlado

TD2. 3: Blueprint (entorno de enseñanza)

TD2. 4: Modelo conceptual de la base de datos

TD2. 5: Maqueta (wireframe) del entorno de aprendizaje

**Requisito: R4:** Configurar perfiles de estudiantes y profesores

**Tareas de Desarrollo:**

TD3. 1: Modelo de contenido de estudiantes e instructores

TD3. 2: Diagrama de clases de estudiantes e instructores

TD3. 3: Maqueta (wireframe) para gestión de perfiles

TD3. 4: Programación de algoritmo para gestión de perfiles

**Producto de Verificación 23:** Informe de las tareas de desarrollo de los requisitos del Sprint.

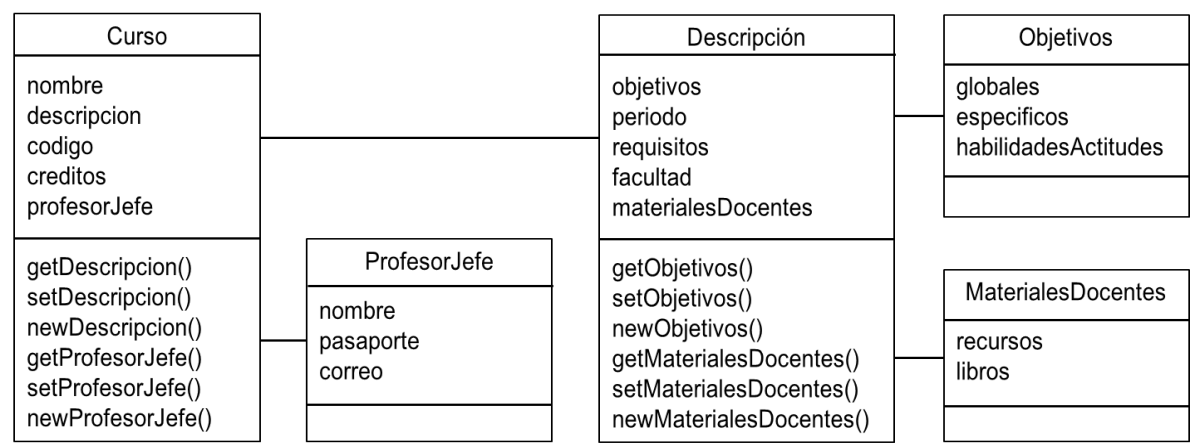
## P24. Diagrama de Clases UML

**Autor:** Arquitecto de la Información (L.R.)

**Fecha:** 27 Julio 2016

**UML1:** Diagrama de clases UML de los elementos relacionados con los cursos

**Descripción:** Se describen las clases de los elementos de contenido de los que se componen los cursos ofrecidos por la universidad.



**Producto de Verificación 24:** Diagrama de Clases UML.



#### 4.4.12.2. Tarea 3.4.2. Identificar técnicas ágiles de la Arquitectura de la Información para las tareas

**Descripción:** El objetivo de esta tarea es apoyar el desarrollo de las tareas relacionadas con la AI a través de la identificación de técnicas ágiles para su desarrollo.

En la realización de la Tarea 3.4.2, también se utilizará el módulo “Gestionar Tareas (Mod-42)”. Este módulo permite asociar técnicas ágiles de la AI a las tareas de desarrollo identificadas. Las opciones para configurar las técnicas ágiles de la AI se encuentran en el apartado Incorporate Information Architecture Agile Techniques del módulo (ver Figura 64). En este caso, la Figura 64 muestra la asignación de la técnica InterArch-T (ver campo Added IA Techniques) a la tarea de desarrollo Modelo de contenido de estudiantes e instructores (ver campo Task) del requisito “R4: Configurar perfiles de estudiantes y profesores” (ver campo Backlog Item).

**Información de entrada:** Esta tarea utiliza como fuentes externas un conjunto de técnicas ágiles para integrar la AI (E9) y el Informe de las tareas de desarrollo de los requisitos del Sprint (P23) de la Tarea 3.4.1

**Producto de salida:** Esta tarea genera, como principal salida, el Informe de las técnicas ágiles de la Arquitectura de la Información relacionadas con las tareas de desarrollo (P25). Al igual que la Tarea 3.4.1, este producto también se puede consultar con el módulo “Gestionar Tareas (Mod-42)” (ver Figura 64) y se presenta en detalle a continuación:

P25. Informe de las técnicas ágiles de la Arquitectura de la Información relacionadas con las tareas de desarrollo	
Autor: Equipo de Desarrollo (M.R. y A.G.)	
Fecha: 01 Agosto 2016	
S1: Sprint 1	
Requisito: R1: Crear y editar cursos	
Tareas de Desarrollo	Técnica ágil de la AI
TD1. 1: Maqueta (wireframe) para crear cursos	InterArch-T, Lo-fi prototyping
TD1. 2: Diagrama de clases de cursos	InterArch-T
Requisito: R2: Gestionar cursos en un entorno integrado	
Tareas de Desarrollo	Técnica ágil de la AI
TD2. 1: Inventario de contenido (Estudiante, Curso, Instructores)	Background investigation, Interviews, Meetings

TD2. 2: Vocabulario controlado	Consistency inspection, Participatory design
TD2. 3: Blueprint (entorno de enseñanza)	Interface design patterns, Lo-fi prototyping, Participatory design
TD2. 5: Maqueta (wireframe) del entorno de aprendizaje	Diagramming, Lo-fi prototyping
<b>Requisito: R4: Configurar perfiles de estudiantes y profesores</b>	
<b>Tareas de Desarrollo</b>	<b>Técnica ágil de la AI</b>
TD3. 1: Modelo de contenido de estudiantes e instructores	InterArch-T
TD3. 2: Diagrama de clases de estudiantes e instructores	InterArch-T
TD3. 3: Maqueta (wireframe) para gestión de perfiles	InterArch-T, Lo-fi prototyping

**Producto de Verificación 25:** Informe de las técnicas ágiles de la Arquitectura de la Información relacionadas con las tareas de desarrollo.

#### 4.4.12.3. Tarea 3.4.3. Estimar duración de tareas

**Descripción:** El objetivo de esta tarea es determinar la duración de las tareas de desarrollo definidas para cada requisito.

Al igual que en las tareas 3.4.1 y 3.4.2, en la realización de esta tarea también se utilizará el módulo “Gestionar Tareas (Mod-42)” que permite registrar la estimación de las tareas de desarrollo identificadas (en días, horas, minutos). Como se puede observar en la Figura 64, la tarea de desarrollo Modelo de contenido de estudiantes e instructores tiene una estimación de 6 horas (ver campo Estimate).

**Información de entrada:** Esta tarea utiliza como principal fuente de entrada el Informe de las tareas de desarrollo de los requisitos del Sprint (P23) de la Tarea 3.4.1.

**Producto de salida:** La Tarea 3.4.3 genera, como resultado, un Informe sobre la estimación de las tareas de desarrollo (P26). Al igual que en la Tarea 3.4.1 y la Tarea 3.4.2, este producto también se puede consultar con el módulo “Gestionar Tareas (Mod-42)” (ver Figura 64) y se presenta a continuación:

<b>P26. Informe sobre la estimación de las tareas de desarrollo</b>	
<b>Autor:</b> Equipo de Desarrollo (M.R. y A.G.)	
<b>Fecha:</b> 01 Agosto 2016	
<b>S1: Sprint 1</b>	
<b>Requisito:</b> R1: Crear y editar cursos	
<b>Tareas de Desarrollo</b>	<b>Estimación</b>
TD1. 1: Maqueta (wireframe) para crear cursos	8 Horas
TD1. 2: Diagrama de clases de cursos	4 Horas
TD1. 3: Programación de algoritmo para crear cursos	16 Horas
TD1. 4: Algoritmo para registrar actividad de los estudiantes	8 Horas
<b>Requisito:</b> R2: Gestionar cursos en un entorno integrado	
<b>Tareas de Desarrollo</b>	<b>Estimación</b>
TD2. 1: Inventario de contenido (Estudiante, Curso, Instructores)	8 Horas
TD2. 2: Vocabulario controlado	4 Horas
TD2. 3: Blueprint (entorno de enseñanza)	8 Horas
TD2. 4: Modelo conceptual de la base de datos	6 Horas
TD2. 5: Maqueta (wireframe) del entorno de aprendizaje	12 Horas
<b>Requisito:</b> R4: Configurar perfiles de estudiantes y profesores	
<b>Tareas de Desarrollo</b>	<b>Estimación</b>
TD3. 1: Modelo de contenido de estudiantes e instructores	6 Horas
TD3. 2: Diagrama de clases de estudiantes e instructores	4 Horas
TD3. 3: Maqueta (wireframe) para gestión de perfiles	8 Horas
TD3. 4: Programación de algoritmo para gestión de perfiles	12 Horas

**Producto de Verificación 26:** Informe sobre la estimación de las tareas de desarrollo.

#### 4.4.12.4. Tarea 3.4.4. Asignar responsables de tareas

**Descripción:** La última tarea de la actividad 3.4 tiene como objetivo establecer los responsables más idóneos de las tareas de desarrollo, con el fin de alcanzar los requisitos.

En la ejecución de la Tarea 3.4.4, también se utilizará el módulo “Gestionar Tareas (Mod-42)” que permite identificar el responsable de cada tarea de desarrollo. La Figura 64 muestra la asignación del responsable (ver campo *Assigned to*) para la tarea de

desarrollo Modelo de contenido de estudiantes e instructores (ver campo Task).

**Información de entrada:** Esta tarea utiliza, como fuente de información, el Informe sobre la disponibilidad del Equipo de Desarrollo para el Sprint (P19) de la Tarea 3.1.2, y el Informe de las tareas de desarrollo de los requisitos del Sprint (P23) de la Tarea 3.4.1.

**Producto de salida:** La Tarea 3.4.4 tiene como principal resultado el Informe de los responsables de las tareas de desarrollo (P27). Al igual que en la Tarea 3.4.1, la Tarea 3.4.2 y la Tarea 3.4.3, este producto también se puede consultar con el módulo “Gestionar Tareas (Mod-42)” (ver Figura 64) y se presenta en detalle a continuación:

<b>P27. Informe de los responsables de las tareas de desarrollo</b>	
<b>Autor:</b> Equipo de Desarrollo (M.R. y A.G.)	
<b>Fecha:</b> 01 Agosto 2016	
<b>S1: Sprint 1</b>	
<b>Requisito: R1: Crear y editar cursos</b>	
<b>Tareas de Desarrollo</b>	<b>Responsable</b>
TD1. 1: Maqueta (wireframe) para crear cursos	Luis Rojas
TD1. 2: Diagrama de clases de cursos	Mario Reyes
TD1. 3: Programación de algoritmo para crear cursos	Adams Galkus
TD1. 4: Algoritmo para registrar actividad de los estudiantes	Mario Reyes
<b>Requisito: R2: Gestionar cursos en un entorno integrado</b>	
<b>Tareas de Desarrollo</b>	<b>Responsable</b>
TD2. 1: Inventario de contenido (Estudiante, Curso, Instructores)	Luis Rojas
TD2. 2: Vocabulario controlado	Luis Rojas
TD2. 3: Blueprint (entorno de enseñanza)	Luis Rojas
TD2. 4: Modelo conceptual de la base de datos	Adams Galkus
TD2. 5: Maqueta (wireframe) del entorno de aprendizaje	Luis Rojas
<b>Requisito: R4: Configurar perfiles de estudiantes y profesores</b>	
<b>Tareas de Desarrollo</b>	<b>Responsable</b>
TD3. 1: Modelo de contenido de estudiantes e instructores	Luis Rojas
TD3. 2: Diagrama de clases de estudiantes e instructores	Adams Galkus

TD3. 3: Maqueta (wireframe) para gestión de perfiles	Luis Rojas
TD3. 4: Programación de algoritmo para gestión de perfiles	Mario Reyes

**Producto de Verificación 27:** Informe de los responsables de las tareas de desarrollo.

Asimismo, el módulo “Consultar Tareas (Mod-41)” también permite consultar todos los detalles de las tareas de desarrollo definidas durante esta actividad. La Figura 65 muestra la Lista de Pendientes del Sprint para el SI de gestión de aprendizaje, con el detalle de las tareas definidas para cada requisito.

**Scrum-UIA-MAT**  
Scrum-UIA Management Tool

Project | Product Backlog | Sprints | Task Board | Burndown | Set up | Project: Gestión de Aprendizaje

Sprint: Sprint 1 | From Aug 8 2016 To Aug 19 2016 - (10 Workdays)

Backlog Item	Status	Type	Total																														
(ID:62) As a Administrador, I want configurar preferencias de los estudiantes y profesores so that digitalizar los perfiles	Assigned to Sprint	User Story	4 Tasks, 30 Hours																														
<table border="1"> <thead> <tr> <th>Id</th> <th>Name</th> <th>Estimate</th> <th>Status</th> <th>IA Techniques</th> <th>Assigned To</th> </tr> </thead> <tbody> <tr> <td>53</td> <td>Modelo de contenido de estudiantes e instructores</td> <td>6 Hours</td> <td>To do</td> <td>InterArch-T</td> <td>Luis Rojas</td> </tr> <tr> <td>54</td> <td>Diagrama de clases de estudiantes e instructores</td> <td>4 Hours</td> <td>To do</td> <td>InterArch-T</td> <td>Adams Galkus</td> </tr> <tr> <td>55</td> <td>Maqueta (wireframe) para gestión de perfiles</td> <td>8 Hours</td> <td>To do</td> <td>Lo-fi prototyping</td> <td>Luis Rojas</td> </tr> <tr> <td>56</td> <td>Programación de algoritmo para gestión de perfiles</td> <td>12 Hours</td> <td>To do</td> <td></td> <td>Mario Reyes</td> </tr> </tbody> </table>	Id	Name	Estimate	Status	IA Techniques	Assigned To	53	Modelo de contenido de estudiantes e instructores	6 Hours	To do	InterArch-T	Luis Rojas	54	Diagrama de clases de estudiantes e instructores	4 Hours	To do	InterArch-T	Adams Galkus	55	Maqueta (wireframe) para gestión de perfiles	8 Hours	To do	Lo-fi prototyping	Luis Rojas	56	Programación de algoritmo para gestión de perfiles	12 Hours	To do		Mario Reyes			
Id	Name	Estimate	Status	IA Techniques	Assigned To																												
53	Modelo de contenido de estudiantes e instructores	6 Hours	To do	InterArch-T	Luis Rojas																												
54	Diagrama de clases de estudiantes e instructores	4 Hours	To do	InterArch-T	Adams Galkus																												
55	Maqueta (wireframe) para gestión de perfiles	8 Hours	To do	Lo-fi prototyping	Luis Rojas																												
56	Programación de algoritmo para gestión de perfiles	12 Hours	To do		Mario Reyes																												
(ID:60) As a Estudiante, I want un entorno virtual so that la información de aprendizaje pueda ser organizada y configurada	Assigned to Sprint	IAU User Story	5 Tasks, 38 Hours																														
(ID:59) As a Instructor & Administrador, I want crear y editar los cursos so that configurar los planes de estudio	Assigned to Sprint	User Story	4 Tasks, 36 Hours																														

Manage Tasks | Mod-41

All Rights Reserved - Contact Us  
Universidad Autónoma de Madrid-Escuela Politécnica Superior

Search This Site  
Search

Links  
[-Add End-Users to the Project](#)  
[-Add Aspects to the Project](#)  
[-Add Elements to Project's Aspects](#)  
[-Set Up Project's Aspects Priority](#)  
[-Manage Team Capacity](#)

**Figura 65:** Consulta de las tareas de desarrollo definidas para el Sprint inicial del SI de gestión de aprendizaje, utilizando el módulo “Consultar Tareas (Mod-41)”.

#### 4.4.13.Actividad 3.5. Planificación de las tareas de desarrollo

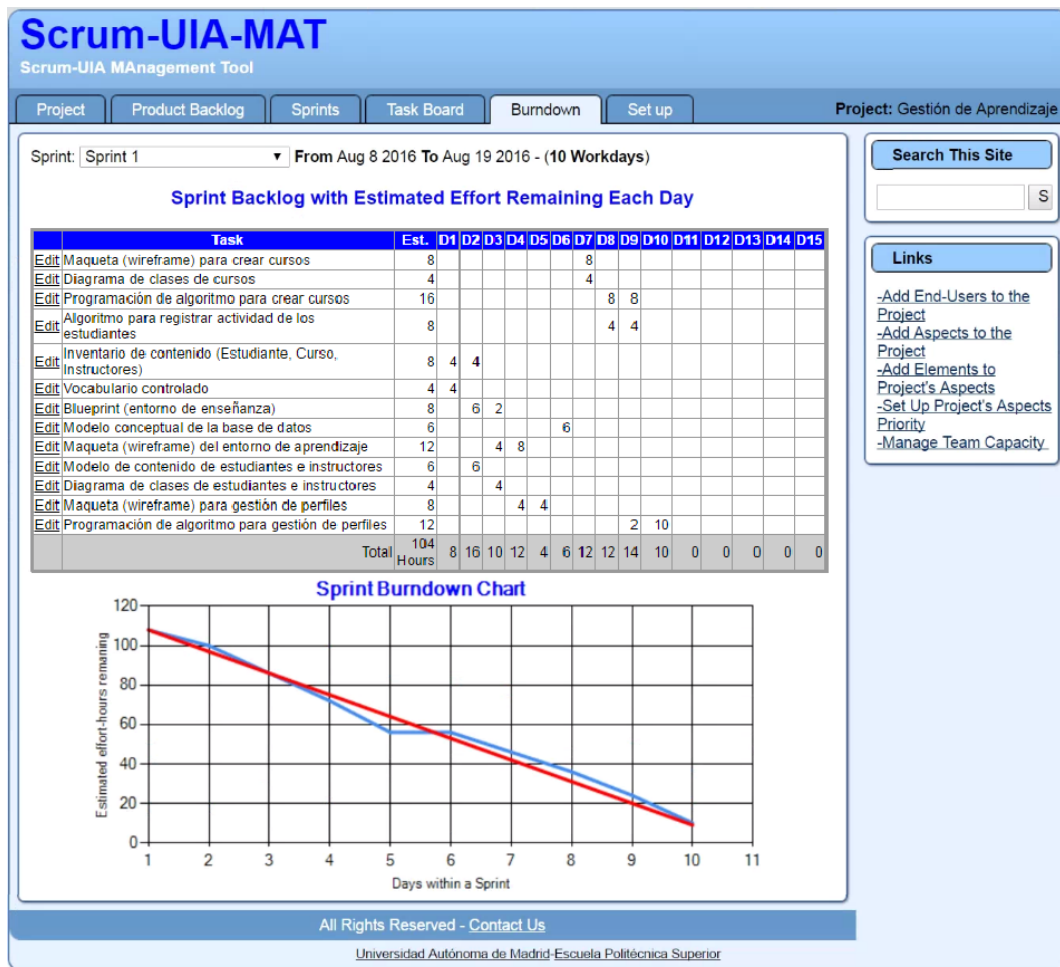
Una vez definidas y descritas todas las tareas necesarias para completar los requisitos del Sprint (Actividad 3.4), se realizará una planificación diaria de las tareas de desarrollo.

##### 4.4.13.1. Tarea 3.5.1. Determinar la planificación de las tareas

**Descripción:** El objetivo de esta tarea es identificar los días en los que se implementarán las tareas.

En la ejecución de la Tarea 3.5.1, se utilizará el módulo “Estimación del Esfuerzo de las Tareas (Mod-51)” de Scrum-UIA-MAT. Este módulo permite determinar el día del Sprint en el que se llevará a cabo cada tarea. En la Figura 66, se presenta la planificación diaria de

las tareas para el primer Sprint del SI de gestión de aprendizaje. Adicionalmente, este módulo también permite visualizar, de manera gráfica, la planificación realizada.



**Figura 66:** Determinación de la planificación de las tareas de desarrollo, mediante el módulo “Estimación del Esfuerzo de las Tareas (Mod-51)”.

**Información de entrada:** Esta tarea utiliza como fuentes de información el Informe de las tareas de desarrollo de los requisitos del Sprint (P23) de la Tarea 3.4.1, el Informe sobre la estimación de las tareas de desarrollo (P26) de la Tarea 3.4.3, y el Informe de los responsables de las tareas de desarrollo (P27) de la Tarea 3.4.4.

**Producto de salida:** La Tarea 3.5.1 tiene como resultado un Diagrama de la planificación diaria de las tareas de desarrollo (P28), que también se puede consultar con el módulo “Estimación del Esfuerzo de las Tareas (Mod-51)” (ver Figura 66). A continuación, se presenta el producto (P28) con la planificación de las tareas de desarrollo para el primer Sprint del SI de gestión de aprendizaje:

<b>P28. Diagrama de la planificación diaria de las tareas de desarrollo</b> <b>Autor:</b> Equipo de Desarrollo (M.R. y A.G.) <b>Fecha:</b> 01 Agosto 2016												
S1: Sprint 1												
Tareas de Desarrollo	Est.	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	
Maqueta (wireframe) para crear cursos	8							8				
Diagrama de clases de cursos	4							4				
Programación de algoritmo para crear cursos	16								8	8		
Algoritmo para registrar actividad de los estudiantes	8								4	4		
Inventario de contenido (Estudiante, Curso, Instructores)	8	4	4									
Vocabulario controlado	4	4										
Blueprint (entorno de enseñanza)	8		6	2								
Modelo conceptual de la base de datos	6						6					
Maqueta (wireframe) del entorno de aprendizaje	12			4	8							
Modelo de contenido de estudiantes e instructores	6		6									
Diagrama de clases de estudiantes e instructores	4			4								
Maqueta (wireframe) para gestión de perfiles	8				4	4						
Programación de algoritmo para gestión de perfiles	12									2	10	
<b>Total</b>	<b>104</b>	<b>8</b>	<b>16</b>	<b>10</b>	<b>12</b>	<b>4</b>	<b>6</b>	<b>12</b>	<b>12</b>	<b>14</b>	<b>10</b>	

**Producto de Verificación 28:** Diagrama de la planificación diaria de las tareas de desarrollo.

Cabe destacar que se ha organizado la ejecución de las tareas de desarrollo, con el fin de fomentar el desarrollo incremental, de acuerdo a los niveles de fidelidad de los entregables de la AI. Por ejemplo, la tarea de desarrollo Modelo de contenido de estudiantes e instructores se ha planificado para elaborarla con la técnica

InterArch-T (ver Producto de Verificación 25) en el segundo día del Sprint (ver Producto de Verificación 28). De este modo, la elaboración de esta tarea hace posible la generación de información de análisis y diseño (diagrama de clases UML en formato XMI), para dar continuidad a la tarea de desarrollo Diagrama de clases de estudiantes e instructores, en base al trabajo inicial del Arquitecto de la Información.

Por otro lado, es importante mencionar que el Producto de Verificación 4 corresponde a un modelo de contenido de los elementos relacionados con los cursos del SI para la gestión de aprendizaje de la universidad. Este producto se ha elaborado en la Tarea 1.2.1 (Investigación Contextual) y se ha creado mediante la herramienta InterArch, lo que permite generar automáticamente un diagrama de clases UML de los elementos relacionados con los cursos (Producto de Verificación 24). Por lo tanto, este diagrama es la base para elaborar la tarea de desarrollo Diagrama de clases de cursos (planificada para el sexto día del Sprint, ver Producto de Verificación 28), facilitando llegar a una solución de diseño, más cercana al dominio de la solución, de la que se benefician los miembros del Equipo de Desarrollo, a partir de las representaciones conceptuales que crea el Arquitecto de la Información.

#### **4.4.14.Actividad 4.1. Inspección de tareas de desarrollo**

Una vez comenzada la Ejecución del Sprint, así como durante todos los días del Sprint, se inspeccionarán las tareas de desarrollo, con el fin de revisar sus estados y verificar el cumplimiento de los criterios de aceptación de los usuarios finales.

##### **4.4.14.1. Tarea 4.1.1. Examinar las tareas de desarrollo**

**Descripción:** El objetivo de esta tarea es revisar las tareas de desarrollo y los retos que tienen los miembros del Equipo de Desarrollo.

**Información de entrada:** Esta tarea utiliza como fuentes de información la Lista del Producto con los elementos descritos en formato de Historia de Usuario (P13) de la Tarea 2.2.2, el Informe de las tareas de desarrollo de los requisitos del Sprint (P23), proporcionado por la Tarea 3.4.1, el Informe de los responsables de las tareas de desarrollo (P27), proporcionado por la Tarea 3.4.4, el Informe de las técnicas ágiles de la Arquitectura de la Información relacionadas con las tareas de desarrollo (P25) de la Tarea 3.4.2, y el Diagrama de la planificación diaria de las tareas de desarrollo (P28) de la Tarea 3.5.1.

**Producto de salida:** A continuación, se presenta el Informe de la revisión de las tareas de desarrollo (P29) que se generó en la Tarea 4.1.1.



<b>P29. Informe de la revisión de las tareas de desarrollo</b> <b>Autor:</b> Equipo de Desarrollo (M.R. y A.G.) <b>Fecha:</b> 10 Agosto 2016	
<b>S1: Sprint 1</b> <b>Día del Sprint:</b> Día 3 del Sprint (Miércoles 10 de agosto de 2016)	
<b>Tareas de Desarrollo</b>	<b>Observaciones</b>
TD1. 1: Maqueta (wireframe) para crear cursos	Tarea no comenzada. Planificada para el día 7 del Sprint. La maqueta debe soportar los elementos definidos en el diagrama de clases (TD1. 2).
TD1. 2: Diagrama de clases de cursos	Tarea no comenzada. Planificada para el día 6 del Sprint. Se debe utilizar, como base, el resultado del Producto de Verificación 24.
TD1. 3: Programación de algoritmo para crear cursos	Tarea no comenzada. Planificada para los días 8 y 9 del Sprint.
TD1. 4: Algoritmo para registrar actividad de los estudiantes	Tarea no comenzada. Planificada para los días 8 y 9 del Sprint.
TD2. 1: Inventario de contenido (Estudiante, Curso, Instructores)	Tarea finalizada. Inventario de contenido identifica los datos de preferencias de los estudiantes y profesores.
TD2. 2: Vocabulario controlado	Tarea finalizada. Vocabulario controlado identifica las diferentes expresiones, utilizadas por los estudiantes y profesores, relacionadas con los cursos.
TD2. 3: Blueprint (entorno de enseñanza)	Tarea en desarrollo. Blueprint identifica las opciones para adaptarse a diferentes tipos de usuarios.
TD2. 4: Modelo conceptual de la base de datos	Tarea no comenzada. Planificada para el día 6 del Sprint.
TD2. 5: Maqueta (wireframe) del entorno de aprendizaje	Tarea en desarrollo. La maqueta debe soportar las opciones de adaptación, identificadas en la tarea TD2. 3.

TD3. 1: Modelo de contenido de estudiantes e instructores	Tarea finalizada. El modelo de contenido es desarrollado (Arquitecto de la Información) con InterArch, y considera los datos de los estudiantes e instructores.
TD3. 2: Diagrama de clases de estudiantes e instructores	Tarea en desarrollo. Se utiliza, como base, el resultado de la tarea TD3. 1.
TD3. 3: Maqueta (wireframe) para gestión de perfiles	Tarea no comenzada. Planificada para los días 4 y 5 del Sprint.
TD3. 4: Programación de algoritmo para gestión de perfiles	Tarea no comenzada. Planificada para los días 9 y 10 del Sprint.

**Producto de Verificación 29:** Informe de la revisión de las tareas de desarrollo.

#### 4.4.15. Actividad 4.2 Revisión del incremento del producto

Una vez finalizada la Ejecución del Sprint, en esta actividad se evaluará el incremento del producto a través de la participación directa de los usuarios finales.

##### 4.4.15.1. Tarea 4.2.1. Examinar el incremento potencial del producto

**Descripción:** El objetivo de esta tarea es evaluar el incremento potencial del producto.

**Información de entrada:** Esta tarea utiliza como fuentes de información la Lista del Producto con los elementos descritos en formato de Historia de Usuario (P13) de la Tarea 2.2.2, el Informe de las tareas de desarrollo de los requisitos del Sprint (P23), proporcionado por la Tarea 3.4.1, y el Informe de la revisión de las tareas de desarrollo (P29) de la Tarea 4.1.1.

**Producto de salida:** A continuación, se presenta el Informe sobre la revisión del incremento del producto (P30) que se generó en la Tarea 4.2.1.

<b>P30. Informe sobre la revisión del incremento del producto</b>	
<b>Autor:</b> Equipo de Desarrollo (M.R. y A.G.)	
<b>Fecha:</b> 22 Agosto 2016	
<b>S1:</b> Sprint 1	
<p><b>Observaciones:</b> Un estudiante de la universidad ha participado para revisar el diseño del entorno Web que permite gestionar y configurar los cursos. El estudiante se ha manifestado conforme con la composición de los elementos estructurales del contenido, pero ha indicado la necesidad de que el SI tenga una cuenta de usuario, integrada con los demás servicios ofrecidos por la universidad (autenticación en los ordenadores, VPN, correo electrónico y biblioteca).</p>	

Asimismo, un trabajador de la universidad (del área administrativa) ha participado para validar las funcionalidades que permiten configurar las preferencias de los estudiantes e instructores. En este caso, el personal administrativo ha señalado que los datos recopilados permiten identificar las prioridades más relevantes de los estudiantes y profesores. No obstante, se han hecho sugerencias para que estas funcionalidades se extiendan a los estudiantes, con el fin de que éstos puedan autoconfigurar sus preferencias.

**Producto de Verificación 30:** Informe sobre la revisión del incremento del producto.

#### 4.4.16. Actividad 4.3 Inspección de prácticas de trabajo

Una vez llevada a cabo la evaluación del incremento potencial del producto (Actividad 4.3), en esta actividad se analizará el Sprint completado, con el fin de indagar mejoras para el siguiente Sprint.

##### 4.4.16.1. Tarea 4.3.1. Examinar el Sprint completado

**Descripción:** El objetivo de esta tarea es inspeccionar el Sprint llevado a cabo, con el fin de identificar las mejoras para el siguiente Sprint.

**Información de entrada:** Esta tarea utiliza como fuentes de información el Informe de las técnicas ágiles de la Arquitectura de la Información relacionadas con las tareas de desarrollo (P25) de la Tarea 3.4.2, y el Informe sobre la revisión del incremento del producto (P30) de la Tarea 4.2.1.

**Producto de salida:** A continuación, se presenta el Informe sobre la inspección de las prácticas de trabajo (P31) que se generó en la Tarea 4.2.1.

P31. Informe sobre la inspección de las prácticas de trabajo	
<b>Autor:</b> Equipo de Desarrollo (M.R. y A.G.)	
<b>Fecha:</b> 22 Agosto 2016	
S1: Sprint 1	
<b>Observaciones:</b> El Equipo de Desarrollo ha manifestado la necesidad de incorporar elementos cualitativos en los modelos de contenido, elaborados por el Arquitecto de la Información, con el fin de que se incorpore información sobre los criterios de aceptación de los usuarios finales, para que los miembros del Equipo de Desarrollo puedan analizarla.	

**Producto de Verificación 31:** Informe sobre la inspección de las prácticas de trabajo.

#### 4.5. Recapitulación

A lo largo de este capítulo, se ha presentado una verificación de las tareas y de las actividades de la metodología Scrum-UIA, con el soporte de la herramienta Scrum-UIA-

MAT, a través de la aplicación de un ejemplo práctico de desarrollo. Esta verificación consistió en la ejecución, de manera ordenada, de todas las tareas y actividades de Scrum-UIA, así como de los módulos de Scrum-UIA-MAT que las soportaban, considerando como fuente del dominio del problema la construcción de un SI de gestión de aprendizaje para una universidad.

La ejecución de las actividades se inició con el Grupo de Actividades de la Investigación Contextual, donde se han discutido y descrito las características principales de los usuarios finales (estudiantes, instructores, observadores y administradores), con el fin de considerar su perspectiva durante todo el proceso de Scrum-UIA. Asimismo, se han identificado formalmente las prioridades del proyecto, mediante el uso de elementos y aspectos relevantes, relacionados con estas prioridades. Además, se ha generado un modelo de contenido a través del uso de la herramienta InterArch que permite obtener de manera automática un diagrama de clases UML, que se puede integrar en fases más cercanas del dominio de la solución. Los resultados de estas actividades permiten proveer un panorama general del proyecto, con una visión global con respecto a las preferencias de la usabilidad, de la AI y del valor de negocio.

Posteriormente, se han ejecutado las tareas del Grupo de Actividades de la Gestión de Requisitos que permitieron identificar, analizar, priorizar (QMSPR) y estimar los requisitos del proyecto. Estas actividades se han conducido de acuerdo a las preferencias, identificadas en el Grupo de Actividades de la Investigación Contextual, con el fin de promover la implementación de los requisitos a través de las prioridades de la usabilidad, de la AI y del valor de negocio, de una manera ágil y centrada en el usuario. Asimismo, en los resultados de estas actividades, también se provee una definición de “Terminado” para los requisitos del siguiente Sprint, que facilita obtener un entendimiento común sobre su alcance y demandas.

Una vez descritos los requisitos del proyecto, se ejecutaron las tareas del Grupo de Actividades de la Planificación del Sprint. En este grupo de actividades, se identificaron, estimaron y planificaron las tareas de desarrollo necesarias para implementar los requisitos para el primer Sprint del proyecto. De esta manera, se han establecido las tareas de desarrollo relacionadas con la AI, para confeccionarlas con la técnica InterArch-T, y planificadas para fomentar, durante la Ejecución del Sprint, el desarrollo de forma incremental a través de los diferentes niveles de fidelidad de los entregables de la AI.

Por último, se han ejecutado las tareas del Grupo de Actividades de Inspección y Mejora Continua. En estas actividades, se han inspeccionado y evaluado las tareas de desarrollo y las prácticas de trabajo del primer Sprint del proyecto, con el fin de fomentar el Diseño Centrado en el Usuario de las tareas y la implicación activa de los usuarios finales.

Adicionalmente, en la ejecución de las distintas tareas de Scrum-UIA, se han utilizado los módulos de la herramienta Scrum-UIA-MAT que las soportan. Esta herramienta permite implementar el enfoque propuesto en el método de priorización QMPSR, y considera

diferentes funcionalidades para apoyar el Diseño Centrado en el Usuario y el desarrollo ágil de la AI.

Finalmente, la realización de esta verificación de Scrum-UIA, junto con el soporte de InterArch-T y Scrum-UIA-MAT, permite ilustrar y aclarar, a través de un ejemplo concreto, la idoneidad de las tareas, las actividades y los productos propuestos para integrar la AI dentro de un modelo de proceso ágil de desarrollo centrado en el usuario, y corroborar la hipótesis de partida **H2.1**: Es posible idear una metodología para integrar la Arquitectura de la Información en un proceso ágil de desarrollo centrado en el usuario.

No obstante, se hace también necesario evaluar las características y las funcionalidades de los diferentes componentes propuestos (QMPSR, InterArch y Scrum-UIA-MAT) para promover el desarrollo ágil y centrado en el usuario en Scrum-UIA. Por lo tanto, en el siguiente capítulo, se realizan distintos experimentos para evaluar QMPSR, InterArch y Scrum-UIA-MAT, y corroborar así las hipótesis de partida restantes (**H2.2**, **H2.3** y **H2.4**).



# Capítulo 5: Evaluación

---

**E**n este capítulo, se detallan diferentes experimentos para validar y evaluar la solución completa. Más específicamente, se describen los experimentos ejecutados para evaluar la eficiencia de las funcionalidades y características de los tres componentes propuestos (QMPSR, InterArch y Scrum-UIA-MAT) en la metodología Scrum-UIA, que facilitan la gestión de los requisitos en entornos cambiantes, considerando las capacidades y exigencias de los usuarios finales durante todo el ciclo de vida del desarrollo de software. En primer lugar, en la sección 5.1, se detallan los nueve experimentos ejecutados para evaluar QMPSR frente a otros métodos de priorización existentes, con el fin de corroborar la hipótesis de partida **H2.2**. Posteriormente, en la sección 5.2, se describe la evaluación realizada con 12 Arquitectos de la Información para medir la usabilidad de InterArch, y corroborar la hipótesis de partida **H2.3**. Finalmente, en la sección 5.3, se presenta la experimentación llevada a cabo con 16 Ingenieros del Software para medir el grado de usabilidad de Scrum-UIA-MAT, y corroborar la última hipótesis de partida de esta investigación **H2.4**.

## 5.1. Evaluación de las Colisiones de Requisitos en QMPSR

En esta sección, se describen los experimentos realizados para evaluar el método de priorización QMPSR, frente a seis métodos de priorización existentes, con el objetivo de examinar su flexibilidad para gestionar el proceso de priorización en entornos complejos, y corroborar la hipótesis de partida **H2.2**: Es posible idear un método de priorización que permita a la metodología anteriormente ideada (hipótesis de partida **H2.1**) conducir las prioridades de los requisitos a través de las preferencias de la usabilidad, de la Arquitectura de la Información y del valor de negocio, generando una clasificación final que permita reducir las colisiones de requisitos y presentar un buen comportamiento en términos de escalabilidad. Para la evaluación del método de priorización QMPSR, se ha creado un marco comparativo que permite realizar evaluaciones con diferentes métodos de priorización, en base a características específicas. En primer lugar, en la sección 5.1.1, se describen los diferentes arreglos que se pueden configurar con el marco comparativo propuesto. Posteriormente, en la sección 5.1.2, se describen el criterio de medición, los parámetros de configuración, las variables, las preguntas de investigación y los métodos ejecutados en los experimentos. En la sección 5.1.3, se describen los tres pasos del proceso que permiten ejecutar los experimentos. Finalmente, en las secciones 5.1.4 y 5.1.5, se presentan los resultados y las discusiones de los diferentes experimentos ejecutados, respectivamente.

### 5.1.1. Marco Comparativo para la Evaluación Empírica de los Métodos de Priorización

La heterogeneidad de los métodos de priorización, en relación con el tipo de configuración requerida tanto para la ejecución como para la evaluación, hace que sea difícil contrastar y evaluar la misma característica en diferentes propuestas. Un enfoque para abordar y manejar tal inconveniente es identificar y definir un marco comparativo, para determinar similitudes y patrones comunes entre los métodos de priorización. De hecho, es esencial identificar escenarios de disposición, para llevar a cabo experimentos con diferentes propuestas en condiciones similares.

Asumiendo este desafío, se ha creado un marco para llevar a cabo un análisis comparativo de diferentes métodos de priorización, en base a características específicas, con el fin de estudiar la idoneidad de QMPSR para adaptarse, de manera flexible, a los entornos dinámicos, y conducir eficientemente el proceso de priorización en Scrum-UIA. Por lo tanto, se ha seleccionado y evaluado una serie de métodos de priorización, con el objetivo de observar y comparar las colisiones de requisitos, generadas en cada uno de ellos. Como se mencionó anteriormente, una colisión de requisitos corresponde a la situación, en la que dos o más requisitos tienen el mismo valor de priorización en una clasificación final. Aunque este criterio de medición (colisiones de requisitos) no se ha considerado en los estudios previos de investigación (Berander et al., 2006), se hace relevante en procesos dinámicos de priorización, debido a varios motivos. Por ejemplo, un gran número de colisiones de requisitos resulta en un aumento del tiempo de toma de decisiones, y reduce la capacidad para discernir el nivel de relevancia de los requisitos. Por otra parte, un número moderado de las colisiones de requisitos permite tener un buen comportamiento en términos de escalabilidad, garantizar la coherencia de las evaluaciones de los requisitos y verificar la exactitud de la clasificación final mientras el número de requisitos aumenta.

En concreto, se han considerado los siguientes métodos de priorización, presentados en la sección 2.5.2, con el fin de analizar las colisiones: QMPSR, MoSCoW (Stapleton, 1997), Orientado al Valor (Azar et al., 2007), Weiger (Wieggers, 1999), Definición de Producto (Fraser, 2002), AHP (Saaty, 1980) y Kano (Kano et al., 1984).

En el diseño del marco comparativo, se han seguido las siguientes pautas generales:

**Una Definición Común del Esfuerzo de Priorización:** Los métodos de priorización utilizan diferentes concepciones y juicios sobre cómo la priorización de los requisitos se debe llevar a cabo. En consecuencia, el primer paso necesario para obtener un marco comparativo es la creación de una definición común de lo que se entiende como el esfuerzo de priorización. Firesmith (Firesmith, 2004) argumenta que las dimensiones de priorización son los factores que influyen en la prioridad de un requisito, y de acuerdo con Berander y Andrews (Berander y Andrews, 2005), los requisitos se pueden priorizar a lo largo de muchas, diferentes, relacionadas e incluso opuestas dimensiones de priorización (por ejemplo, *la importancia, el valor de negocio, sanción, coste, tiempo y riesgo*). Por lo tanto,



el esfuerzo de priorización se considera como el nivel de relevancia de un requisito en una dimensión de priorización particular. En el marco propuesto, el *Número de Dimensión de Priorización* (NPD<sup>19</sup>) se define como:  $NPD = \{1, \dots, \iota\}$ , es decir, un conjunto finito de dimensiones de priorización, de tal manera que  $\iota$  es el número total de las dimensiones de priorización  $\{\iota \in \mathbb{N}: \iota \geq 1\}$ .

**Coordinación del Esfuerzo de Priorización:** Junto con la obtención de un entendimiento común de lo que es el esfuerzo de priorización, también se requiere llevar a cabo diferentes experimentos con métodos, utilizando el mismo nivel de esfuerzo de priorización, con el fin de obtener resultados comparables. Por lo tanto, el segundo paso necesario para componer un marco comparativo consiste en tener una generación coordinada del esfuerzo de priorización entre los diferentes métodos.

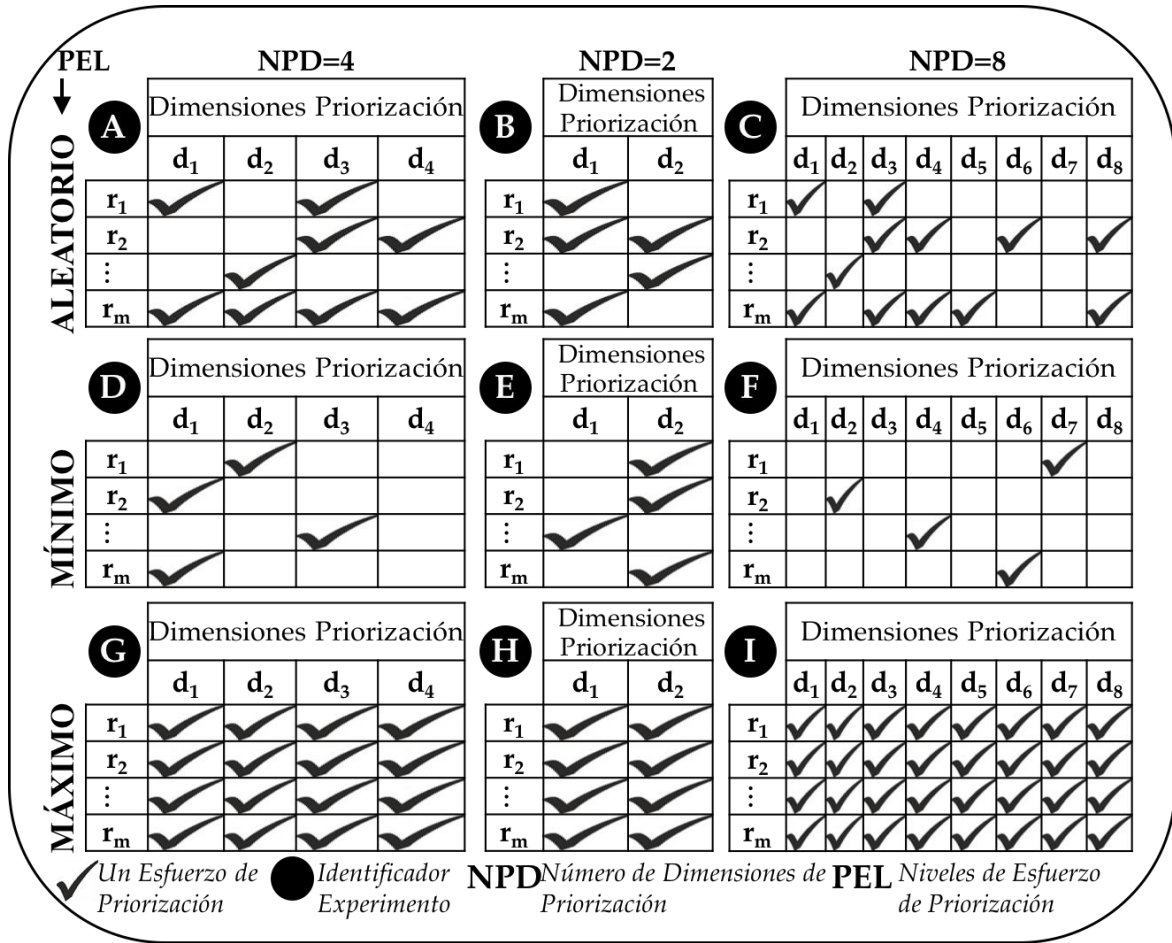
**Definición de Niveles de Esfuerzo de Priorización:** En el último paso necesario para componer un marco comparativo, se definen diferentes *Niveles de Esfuerzo de Priorización* (PEL<sup>20</sup>), con el objetivo de llevar a cabo experimentos con métodos de priorización en diferentes condiciones de rendimiento. El valor de *PEL* permite identificar y seleccionar el número de las dimensiones de priorización que se utilizan para priorizar los requisitos. En el marco propuesto, *PEL* se puede definir de la siguiente manera:  $PEL = \{aleatorio, mínimo, máximo\}$ , es decir, una colección finita de las modalidades de priorización. De esta manera, para un *PEL aleatorio*, el *NPD* para cada uno de los requisitos se identifica aleatoriamente (entre 1 y el valor total de *NPD*, definido para cada experimento). Para un *PEL mínimo*, sólo una de las dimensiones de priorización definidas para cada experimento se identifica aleatoriamente. Finalmente, para un *PEL máximo*, todas las dimensiones de priorización definidas se seleccionan para cada requisito.

Estas directrices generales permiten organizar los diferentes escenarios experimentales para cada método de priorización de manera coordinada. La Figura 67 presenta los diferentes arreglos que se pueden configurar con el marco comparativo propuesto.

---

<sup>19</sup> Por sus siglas del inglés: **N**umber of **P**rioritization **D**imensions.

<sup>20</sup> Por sus siglas del inglés: **P**rioritization **E**ffort **L**evels.



**Figura 67:** Diferentes arreglos para los experimentos en términos de PEL y NPD.

Como se puede ver en la Figura 67, se han creado nueve escenarios experimentales diferentes, variando el *PEL* (*aleatorio*, *mínimo* y *máximo*) y el *NPD* (2, 4 y 8). Por lo tanto, cada experimento se identifica por el *PEL* y *NPD* utilizados, junto con un identificador único (una letra mayúscula entre A e I). Ejemplos de *PEL aleatorio* se pueden observar en los experimentos A, B y C, que aparecen en la Figura 67, donde se definen las dimensiones de priorización de 4, 2 y 8, respectivamente. Del mismo modo, los ejemplos del *PEL mínimo* se pueden observar en los experimentos D, E y F, que aparecen en la Figura 67, donde se definen las dimensiones de priorización de 4, 2 y 8, respectivamente. Finalmente, los experimentos G, H e I, que aparecen en la Figura 67, presentan ejemplos de *PEL máximo*, utilizando las dimensiones de priorización de 4, 2 y 8, respectivamente. Estos experimentos se explican en detalle en las siguientes secciones.

### 5.1.2. Marco Experimental

Utilizando el marco comparativo propuesto anteriormente, nueve experimentos se llevaron a cabo para evaluar los métodos de priorización, con el fin de investigar las colisiones generadas por cada uno de ellos en diferentes condiciones de rendimiento.

Esta evaluación de las colisiones de requisitos se hace relevante en los ambientes ágiles, ya que un gran número de requisitos colisionados afecta negativamente al proceso de priorización de los requisitos, imposibilitando establecer diferencias para priorizar y comprobar, de forma dinámica, la precisión del ranking final de los requisitos.

Por otro lado, es posible identificar los requisitos colisionados de la siguiente manera:

Dada una clasificación final para un conjunto de requisitos  $R$ , llamado  $FR$ , como se define en (12), la colisión del requisito  $r_i$  (donde  $r_i \in R$ ) se define formalmente en términos de la función  $K(r_i)$ , donde  $K: R \rightarrow \{0,1\}$ , con la siguiente interpretación de los valores:

$$K(r_i) = \begin{cases} 1, & \text{si } \exists r_j \neq r_i, r_j \in R, FR(r_j) = FR(r_i); \\ 0, & \text{en caso contrario.} \end{cases} \quad (13)$$

De esta manera, el número total de los requisitos colisionados en una clasificación final particular se define como:

$$TK = \sum_{i=1}^m K(r_i), \quad (14)$$

donde  $K(r_i)$  define la función de colisión para el requisito  $r_i$ , tal y como se describe en la fórmula (13), y  $m$  representa el número total de los requisitos.

#### 5.1.2.1. Variables y Preguntas de Investigación

Una vez descrito formalmente el criterio de medición utilizado en los experimentos, en esta sección se describen las variables y preguntas de investigación consideradas en cada experimento.

Las variables independientes consideradas para cada experimento fueron *PEL* (*aleatorio*, *mínimo* o *máximo*), *NPD* (2, 4, 8) y los conjuntos de entrada de requisitos sintéticos (25, 50, 75, 100, 125, 150, 175 y 200). La variable dependiente considerada para cada experimento fue el número total de los requisitos colisionados en la clasificación final de los métodos de priorización, calculados a través de la fórmula (14).

Por otro lado, las preguntas de investigación para los experimentos se definen en términos de la variable dependiente, y se responden a través de los resultados obtenidos. A continuación, se presentan las preguntas de investigación (descritas anteriormente en la sección 1.2):

**PI15:** ¿Es posible reducir las colisiones de requisitos a través del método de priorización propuesto (QMPSR) dirigido por elementos cualitativos?

**PI16:** ¿En qué medida QMPSR supera a otros métodos de priorización en términos de colisiones de requisitos y de esfuerzo de priorización?

**PI17:** ¿Cómo se comporta el método QMPSR en términos de escalabilidad, en función del número de colisiones de requisitos observadas, cuando el conjunto de requisitos aumenta en tamaño?

### 5.1.2.2. Parámetros de Configuración para la Evaluación de los Métodos de Priorización

En esta sección se describen las características principales de los métodos de priorización seleccionados para evaluar a través de los experimentos. En la Tabla 26, se presentan algunas características de los métodos de priorización. La descripción de estas características permite identificar los escenarios en los que los métodos de priorización se pueden llevar cabo (ver sección 5.1.2.3), y determinar la configuración necesaria para su ejecución en los experimentos (ver sección 5.1.3). A continuación, se describe cada una de las columnas de la Tabla 26:

Métodos	Dimensiones de Priorización	Tamaño	NPD Máximo	Tipo de Evaluación de la Dimensión de Priorización
QMPSR	Aspectos	Adaptable	Infinita	Elementos de los Aspectos
MoSCoW	Interesados	Adaptable	Infinita	<i>Debe, debería, podría, no deseado</i> <sup>21</sup>
Orientado al Valor	Actividades principales	Adaptable	Infinita	0-10
Weiger	Beneficio, sanción, coste, riesgo	Semi-adaptable	4	0-9
Definición de Producto	Técnico, creativo, usuario, negocio	Semi-adaptable	4	0-5
AHP	Requisitos	Semi-adaptable	Número total de los requisitos	1-9
Kano	Presente, ausente	No adaptable	2	<i>Requerido, esperado, indiferente, aceptado, no requerido</i> <sup>22</sup>

**Tabla 26:** Resumen de las principales características de los métodos de priorización.

<sup>21</sup> En inglés: Must, should, could, won't.

<sup>22</sup> En inglés: Like, expect, don't care, live with, dislike.

La columna de *Dimensiones de Priorización*, en la Tabla 26, se utiliza para identificar las dimensiones de priorización que caracteriza cada método. Como se puede ver, las dimensiones de priorización, como *aspectos*, *interesados*, *actividades principales* y *requisitos*, corresponden a los que se utilizan en los métodos QMPSR, MoSCoW, Orientado al Valor y AHP, respectivamente. Del mismo modo, *beneficio*, *sanción*, *coste* y *riesgo* corresponden a las 4 dimensiones de priorización que se utilizan en el método Weiger, mientras que *técnico*, *creativo*, *usuario* y *negocio* corresponden a las 4 dimensiones de priorización que se utilizan en el método de Definición de Producto. Finalmente, *presente* y *ausente* corresponden a las 2 dimensiones de priorización que se utilizan en el método Kano.

Asimismo, la columna de *Tamaño*, en la Tabla 26, se utiliza para identificar si los métodos permiten adaptar el *NPD* para priorizar a cada requisito. El valor *adaptable* indica que el método permite adaptar el *NPD*, el valor *semi-adaptable* ilustra que el método define un *NPD* máximo, pero permite adaptar el *NPD* a usar, mientras que el valor *no adaptable* indica que el método define un *NPD* fijo para priorizar a cada requisito. Como se muestra en la Tabla 26, los métodos QMPSR, MoSCoW y Orientado al Valor permiten adaptar el *NPD*. Por otro lado, los métodos Weiger, Definición de Producto y AHP definen un *NPD* máximo (por ejemplo, en el método Weiger, se definen 4 dimensiones de priorización, es decir, *beneficio*, *sanción*, *coste* y *riesgo*). Sin embargo, estos métodos permiten adaptar el *NPD* a utilizar. Por último, el método Kano define un *NPD* fijo para priorizar a cada requisito.

Por otro lado, la columna de *NPD máximo* se utiliza para identificar si los métodos definen un *NPD* máximo. Como se muestra en la Tabla 26, QMPSR, MoSCoW y Orientado al Valor no definen un *NPD* máximo (*infinito*). Por el contrario, el conjunto de requisitos priorizados corresponde al *NPD* máximo que se define en el método AHP (*número total de los requisitos*). Por último, el *NPD* máximo, que se define en los métodos Weiger y Definición de Producto, corresponde a 4, mientras que en el método Kano corresponde a 2.

Por último, la columna de *Tipo de Evaluación de la Dimensión de Priorización* se utiliza para identificar cómo se lleva a cabo el establecimiento de las prioridades en los requisitos de cada método. Como se muestra en la Tabla 26, en los métodos Orientado al Valor, Weiger, Definición de Producto y AHP, se priorizan los requisitos mediante la asignación de una escala numérica en cada una de sus dimensiones de priorización (por ejemplo, el método Orientado al Valor utiliza una escala numérica entre 0 y 10 para priorizar cada dimensión de priorización de *actividad principal*). Por otra parte, en los métodos MoSCoW y Kano, se priorizan los requisitos a través de la asignación de una escala cualitativa en cada una de sus dimensiones de priorización, con el fin de categorizar los requisitos de mayor importancia (por ejemplo, el método MoSCoW utiliza la escala cualitativa *debe*, *debería*, *podría* y *no deseado* para priorizar a cada dimensión de priorización de *interesado*). Por último, en el método QMPSR, se lleva a cabo la priorización a través de la

asociación entre elementos de los *aspectos* relevantes del proyecto (dimensiones de priorización) y los requisitos, tal y como se explicó en la sección 3.3.1.

### 5.1.2.3. Identificación de los Métodos Evaluados

Finalmente, de acuerdo a las características de los métodos de priorización descritas en la sección anterior, es posible identificar los métodos evaluados en cada experimento.

La Tabla 27 identifica los métodos evaluados en cada experimento. Como se puede observar, los métodos QMPSR, MoSCoW y Orientado al Valor se evaluaron en todos los experimentos, debido a su capacidad para adaptar el *NPD* (ver la columna *Tamaño* de la Tabla 26). Por otro lado, los métodos Weiger y Definición de Producto solamente se evaluaron en los experimentos que implican un *NPD* de 2 o 4, es decir, estos métodos no fueron evaluados en los experimentos con un *NPD* más alto que su *NPD* máximo (ver la columna *NPD Máximo* en la Tabla 26). Del mismo modo, el método AHP solamente se evaluó en los experimentos A, C, G, H e I, ya que permite semi-adaptar el *NPD* a usar (ver la columna *Tamaño* en la Tabla 26). De esta manera, el método AHP se evaluó en los experimentos, donde el esfuerzo total de priorización, calculado con la fórmula (16), fue dos veces mayor que el número de requisitos priorizados. Finalmente, el método Kano solamente se evaluó en el experimento H, ya que este método define un *NPD* fijo (*presente* y *ausente*) para priorizar cada requisito (ver la columna de *Dimensiones de Priorización* de la Tabla 26).

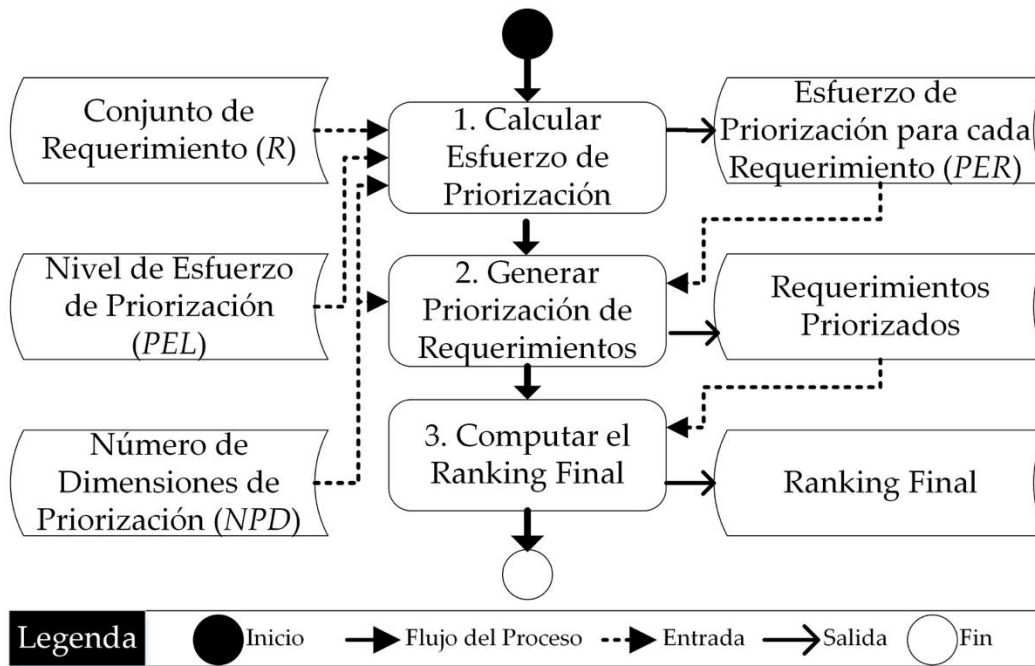
Métodos	Identificador del Experimento								
	A	B	C	D	E	F	G	H	I
QMPSR	x	x	x	x	x	x	x	x	x
Orientado al Valor	x	x	x	x	x	x	x	x	x
MoSCoW	x	x	x	x	x	x	x	x	x
Weiger	x	x		x	x		x	x	
Definición de Producto	x	x		x	x		x	x	
Kano								x	
AHP	x		x				x		x

**Tabla 27:** Identificación de los métodos evaluados en cada experimento.

### 5.1.3. Ejecución de los Experimentos

Una vez descritas las características principales de los experimentos, en esta sección se detallan los pasos realizados en la ejecución de los experimentos.

Los experimentos se llevaron a cabo aplicando el Proceso de Ejecución de Experimento, ilustrado en la Figura 68. Los datos de entrada para cada experimento fueron un valor para *PEL* (*aleatorio*, *mínimo* o *máximo*), un valor para *NPD* (2, 4 o 8) y diferentes conjuntos de requisitos sintéticos (25, 50, 75, 100, 125, 150, 175 y 200). El resultado fue la clasificación final de los requisitos para cada método de priorización. Por ejemplo, todos los métodos involucrados en el experimento A (ver Tabla 27) llevaron a cabo el Proceso de Ejecución de Experimento con *PEL=aleatorio* y *NPD=4* para cada conjunto de requisitos. A continuación, se describe cada paso del Proceso de Ejecución de Experimento (ver Figura 68):



**Figura 68:** Proceso de Ejecución de Experimento.

**1. Calcular el Esfuerzo de Priorización:** El primer paso del proceso tiene como objetivo establecer, para cada conjunto de requisitos, el *Esfuerzo de Priorización para cada Requisito* ( $PER^{23}$ ), con el fin de producir el mismo nivel de esfuerzo de priorización en todos los métodos evaluados, obteniendo, de este modo, resultados comparables para cada uno.

Para cada conjunto de requisitos, el *PER* se identificó de acuerdo con el *PEL* y el *NPD*, ambos definidos en cada experimento (ver Figura 67). Por lo tanto, teniendo en cuenta un experimento que considera un *PEL* particular y un *NPD*, el esfuerzo de priorización para el requisito  $r_i$  se define formalmente en términos de la función  $PER: R \times NPD \times PEL \rightarrow \mathbb{N}$ , o en términos de la siguiente función:

<sup>23</sup> Por sus siglas en inglés: **P**rioritization **E**ffort for each single **R**equirement.

$$PER(r_i, NPD, PEL) = \begin{cases} \beta(1, NPD), & \text{si } PEL \text{ es aleatorio;} \\ 1, & \text{si } PEL \text{ es mínimo;} \\ NPD, & \text{si } PEL \text{ es máximo.} \end{cases} \quad (15)$$

Donde  $\beta(1, NPD)$  es un número aleatorio en el intervalo  $(\kappa, \eta)$  ( $\kappa \in Z$  y  $\eta \in Q$ ), formalmente definido en términos de la función  $\beta$ , donde  $\beta: Z \times Q \rightarrow \mathbb{N} \cup \{\kappa, \eta \in \mathbb{N}: \kappa \leq \eta\}$ , siendo  $Z$  y  $Q$  colecciones de números naturales ( $Z, Q \subset \mathbb{N}$ ). En concreto, en este caso  $\beta$  identifica un número aleatorio en el rango entre 1 y  $NPD$ . Por lo tanto,  $PER(r_i, NPD, PEL) > PER(r_j, NPD, PEL)$  quiere decir que  $r_i$  involucra un número más alto de dimensiones de priorización seleccionadas que  $r_j$  para determinar su nivel de relevancia. Vale la pena destacar que  $PER$  es el mismo en todos los experimentos con el  $PEL$  máximo y mínimo, es decir,  $PER=PEL$  en experimentos con el  $PEL$  máximo, mientras que  $PER=1$  en experimentos con el  $PEL$  mínimo. A modo de ejemplo, en la Figura 67, el esfuerzo de priorización de los requisitos  $r_1$  y  $r_2$  en el experimento C (donde  $PEL=aleatorio$  y  $NPD=8$ ) es 2 y 4, respectivamente, como resultado de la aplicación de  $PER$  en ambos requisitos.

De esta manera, el esfuerzo total de priorización para un conjunto dado de requisitos se define como:

$$TPER(NPD, PEL) = \sum_{i=1}^m PER(r_i, NPD, PEL), \quad (16)$$

donde  $PER(r_i, NPD, PEL)$  define la función del esfuerzo de priorización para el requisito  $r_i$ , tal y como se define en la fórmula (15), y  $m$  representa el número total de los requisitos, incluidos en un conjunto específico.

**2. Generar Priorización de Requisitos:** El segundo paso en el proceso tiene como objetivo establecer la priorización de requisitos para cada método.

En primer lugar, un número aleatorio de  $PER$ , definido en la fórmula (15), se calcula para cada requisito. Sea  $PD = \{d_1, \dots, d_k, \dots, d_{NPD}\}$  una colección finita de dimensiones de priorización, donde  $d_k \in PD$ , de manera que  $NPD$  es el número total de las dimensiones de priorización, definidas en el experimento. La relación del requisito  $r_i$  y la dimensión de priorización  $d_k$  se define formalmente en términos de la función  $\phi(d_k, r_i)$ , donde  $\phi: PD \times R \rightarrow \{0,1\}$ , con la siguiente interpretación para cada uno de estos valores:

$$\phi(d_k, r_i) = \begin{cases} 1, & \text{si } d_k \text{ está seleccionada para } r_i; \\ 0, & \text{en caso contrario.} \end{cases} \quad (17)$$

De esta manera, dado un experimento que considera un  $PEL$  particular y un  $NPD$ , el número total de las dimensiones de priorización seleccionadas aleatoriamente para el requisito  $r_i$  ( $T\phi$ ) se define como:



$$T\phi(r_i, NPD, PEL) = \begin{cases} \sum_{h=1}^{NPD} \phi(d_h, r_i), & \text{si } PEL \text{ es aleatorio;} \\ 1, & \text{si } PEL \text{ es mínimo;} \\ NPD, & \text{si } PEL \text{ es máximo.} \end{cases} \quad (18)$$

Donde  $\phi(d_h, r_i)$  identifica si la dimensión de priorización  $d_h$  está seleccionada para el requisito  $r_i$ , calculado con la fórmula (17). Además, la función  $T\phi(r_i, NPD, PEL)$  también tiene la siguiente propiedad  $T\phi(r_i, NPD, PEL) = PER(r_i, NPD, PEL)$ . Por lo tanto,  $\forall(r_i, r_j) \in R, T\phi(r_i, NPD, PEL) > T\phi(r_j, NPD, PEL)$  quiere decir que el requisito  $r_i$  tiene un mayor número de dimensiones de priorización seleccionadas que el requisito  $r_j$ .

En segundo lugar, la priorización de requisitos se lleva a cabo asignando una evaluación en cada dimensión de priorización, seleccionada para cada requisito. Por lo tanto, la priorización se lleva a cabo aleatoriamente, de acuerdo con el tipo de evaluación de cada método (ver la columna *Tipo de Evaluación de la Dimensión de Priorización* de la Tabla 26).

Inicialmente, se necesita representar el tipo de evaluación para cada método, con el fin de lograr la priorización de los requisitos. Sea  $S = \{l_1, \dots, l_q, \dots, l_h\}$  una colección finita de valores de una escala de calificación, donde  $l_q \in S$ , de tal modo que  $h$  es el número total de los valores de la escala de calificación considerados. De esta manera, la evaluación del requisito  $r_i$  en la dimensión de priorización  $d_k$ , utilizando la escala de valoración  $S$ , se define formalmente en términos de la función  $\psi(d_k, r_i)$ , donde  $\psi: PD \times R \rightarrow S$ , que es definida como:

$$\psi(d_k, r_i) = \begin{cases} l_{\beta(1, |S|)}, & \text{si } \phi(d_k, r_i) = 1; \\ 0, & \text{en caso contrario.} \end{cases} \quad (19)$$

Donde  $\beta(1, |S|)$  identifica un número aleatorio en el rango entre 1 y el número total de los valores de la escala de calificación considerados ( $|S|$ ),  $l_{\beta(1, |S|)} \in S$ , y  $\phi(d_k, r_i)$  identifica si la dimensión de priorización  $d_k$  está seleccionada para el requisito  $r_i$ , tal y como se define en la fórmula (17). Por lo tanto, si la dimensión de priorización  $d_k$  está seleccionada para el requisito  $r_i$ , es decir,  $\phi(d_k, r_i) = 1$ , entonces la priorización del requisito  $r_i$  en la dimensión de priorización  $d_k$  se lleva a cabo a través de la asignación aleatoria de un valor de escala de calificación ( $l_{\beta(1, |S|)}$ ). De lo contrario, la priorización del requisito  $r_i$  en la dimensión de priorización  $d_k$  no se lleva a cabo, debido a que la dimensión de priorización  $d_k$  no está seleccionada para el requisito  $r_i$ , es decir,  $\phi(d_k, r_i) = 0$ .

Por ejemplo, como se muestra en la Tabla 26, en el método Orientado al Valor, se utilizará una escala numérica entre 0 y 10, es decir,  $S = \{0 \dots 10\}$ , aleatoriamente para cada dimensión de priorización seleccionada (en este caso: *actividad principal*), con el fin de

priorizar cada requisito. En QMPSR, por el contrario, la priorización se lleva a cabo a través de la asociación entre elementos de los *aspectos* y los requisitos. Por lo tanto, QMPSR requiere los siguientes pasos para priorizar cada requisito:

- a) **Identificación del número de elementos de cada aspecto seleccionado:** En este caso,  $PD=A$ , por lo cual, para cada aspecto  $d_k$  ( $d_k \in A$ ), el número de elementos asociados al requisito  $r_i$  se elige aleatoriamente. De esta manera, el número de elementos asociados al aspecto  $d_k$  para el requisito  $r_i$  se define formalmente en términos de la función  $EN(r_i, d_k)$ , donde  $EN: R \times A \rightarrow \mathbb{N}$ , que se define como:

$$EN(r_i, d_k) = \begin{cases} \beta(1, |E_k|), & \text{si } \phi(d_k, r_i) = 1; \\ 0, & \text{en caso contrario.} \end{cases} \quad (20)$$

Donde  $\beta(1, |E_k|)$  identifica un número aleatorio en el rango entre 1 y el número total de elementos del aspecto  $d_k$  ( $|E_k|$ ), y  $\phi(d_k, r_i)$  identifica si el aspecto  $d_k$  está seleccionado para el requisito  $r_i$ , tal y como se define en la fórmula (17). Por lo tanto, si el aspecto  $d_k$  es seleccionado para el requisito  $r_i$ , es decir,  $\phi(d_k, r_i) = 1$ , entonces un número aleatorio de elementos del aspecto  $d_k$ , es decir,  $\beta(1, |E_k|)$ , se identifica para el requisito  $r_i$ . De otro modo, ningún elemento para el aspecto  $d_k$  es identificado para el requisito  $r_i$ . Por lo cual,  $EN(r_i, d_k) > EN(r_i, d_h)$  significa que el aspecto  $d_k$  tiene un número más alto de elementos asociados al requisito  $r_i$  que el aspecto  $d_h$  (donde  $d_h \in A$ ).

- b) **Asociación entre los elementos de un aspecto seleccionado y los requisitos:** Un número aleatorio de  $EN$ , como se define en la fórmula (20), de elementos asociados al aspecto  $d_k$ , se calcula para el requisito  $r_i$ . De esta forma, el número total de elementos del aspecto  $d_k$ , aleatoriamente asociado al requisito  $r_i$ , se define formalmente en términos de la función  $T\tau(r_i, d_k)$ , donde  $T\tau: R \times A \rightarrow \mathbb{N}$ , que se define como:

$$T\tau(r_i, d_k) = \sum_{v=1}^{|E_k|} C(e_{kv}, r_i), \quad (21)$$

donde  $C(e_{kv}, r_i)$  identifica si el elemento  $e_{kv}$  está relacionado con el requisito  $r_i$ , tal y como se define en la fórmula (3), y  $E_k$  es una subcolección finita de todos los elementos relacionados con el aspecto  $a_k$  ( $E_k \subset E$ ). Asimismo,  $|E_k|$  identifica el número total de elementos del aspecto  $d_k$ . La función  $T\tau(r_i, d_k)$  también tiene la siguiente propiedad  $T\tau(r_i, d_k) = EN(r_i, d_k)$ . Finalmente,  $T\tau(r_i, d_k) > T\tau(r_j, d_k)$  significa que el requisito  $r_i$  tiene un número más alto de elementos asociados para el aspecto  $d_k$  que el requisito  $r_j$ .

**3. Computar el Ranking Final:** El último paso en la ejecución del experimento consiste en calcular la clasificación final de los requisitos para cada método de priorización. Cada método aplica su propio procedimiento. Por ejemplo, QMPSR calcula la clasificación final de los requisitos de acuerdo con la fórmula (12).

En general, cada uno de los métodos involucrados fue ejecutado llevando a cabo el Proceso de Ejecución de Experimento 10 veces, con el mismo conjunto de datos. Como resultado, en los métodos incluidos en el experimento A (ver las dos primeras columnas de la Tabla 27) el Proceso de Ejecución de Experimento se realizó 80 veces (10 ejecuciones por cada uno de los 8 conjuntos de requisitos, con un  $PEL=aleatorio$  y  $NPD=4$ ). Por lo tanto, el experimento A generó un número total de 480 ejecuciones del Proceso de Ejecución de Experimento para los seis métodos de priorización considerados.

#### 5.1.4. Resultados

Como resultado de las ejecuciones antes mencionadas, los requisitos colisionados se pueden presentar y analizar. A continuación, se presenta la descripción de los resultados:

En la Tabla 28, se muestra el esfuerzo total de priorización, calculado con la fórmula (16), generado en cada experimento para conjuntos de requisitos de 25, 50, 75, 100, 125, 150, 175 y 200. El esfuerzo total de priorización representa el número de evaluaciones realizadas por cada conjunto de requisitos de entrada, con el fin de determinar su nivel de relevancia.

Experimentos	Número de Requisitos Priorizados							
	25	50	75	100	125	150	175	200
<b>A</b>	65	129	191	246	311	374	435	501
<b>B</b>	38	74	112	149	191	223	264	300
<b>C</b>	112	220	336	458	551	672	781	898
<b>D</b>	25	50	75	100	125	150	175	200
<b>E</b>	25	50	75	100	125	150	175	200
<b>F</b>	25	50	75	100	125	150	175	200
<b>G</b>	100	200	300	400	500	600	700	800
<b>H</b>	50	100	150	200	250	300	350	400
<b>I</b>	200	400	600	800	1000	1200	1400	1600

**Tabla 28:** Identificación del esfuerzo de priorización en cada experimento.

Por un lado, la Figura 69, la Figura 70 y la Figura 71 presentan los resultados obtenidos a través de los experimentos A, B y C, respectivamente. Cada gráfico muestra el número de requisitos colisionados (eje y), calculados a través de la fórmula (14), para los conjuntos de

requisitos priorizados de 25, 50, 75, 100, 125, 150, 175 y 200 (eje x), y calculados como la media de 10 ejecuciones, considerando un *PEL aleatorio* y dimensiones de priorización configuradas con diferentes tamaños (4, 2 y 8 dimensiones de priorización, en los experimentos A, B y C, respectivamente). Además, para el caso del método de priorización propuesto en esta tesis doctoral (QMPSR), se incluyen barras de error  $SD^{24}$  para cada conjunto de requisitos priorizados, con el fin de observar la dispersión obtenida. Asimismo, se utilizó la prueba de Mann-Whitney-Wilcoxon para evaluar la diferencia de las medias entre el número de requisitos colisionados obtenidos por QMPSR y los otros métodos en los experimentos A, B y C. En todos los cálculos, el valor del  $p$  fue  $<0,05$ , lo que indica que las diferencias obtenidas son estadísticamente significativas.

Por otro lado, la Figura 72, la Figura 73 y la Figura 74 presentan los resultados adquiridos a través de los experimentos D, E y F, respectivamente. Cada gráfico muestra el número de requisitos colisionados (eje y), calculados a través de la fórmula (14), para los conjuntos de requisitos priorizados de 25, 50, 75, 100, 125, 150, 175 y 200 (eje x). El número de requisitos colisionados se ha calculado como un promedio de 10 ejecuciones, considerando un *PEL mínimo* y dimensiones de priorización configuradas con diferentes tamaños (4, 2 y 8 dimensiones de priorización, en los experimentos D, E y F, respectivamente). En estos experimentos, la prueba de Mann-Whitney-Wilcoxon no se calculó, porque los métodos generaron un número similar de requisitos colisionados para cada conjunto de requisitos. Además, y para el caso de QMPSR, se han incorporado barras de error  $SD$  para cada conjunto de requisitos priorizados, con el fin de examinar la dispersión.

Finalmente, la Figura 75, la Figura 76 y la Figura 77 ilustran los resultados conseguidos a través de los experimentos G, H e I, respectivamente. Cada gráfico presenta el número de requisitos colisionados (eje y), calculado a través de la fórmula (14), para los conjuntos de requisitos priorizados de 25, 50, 75, 100, 125, 150, 175 y 200 (eje x). El número de requisitos colisionados se calculó como un promedio de 10 ejecuciones, considerando un *PEL máximo* y dimensiones de priorización configuradas con diferentes tamaños (4, 2 y 8 dimensiones de priorización, en los experimentos G, H e I, respectivamente). Por otra parte, para el caso de QMPSR, se incorporaron las barras de error  $SD$  para cada conjunto de requisitos priorizados, con el propósito de examinar la dispersión. Además, se utilizó la prueba de Mann-Whitney-Wilcoxon para evaluar la diferencia de las medias entre el número de requisitos colisionados obtenidos por QMPSR y los otros métodos en los experimentos G, H e I. En todos los cálculos el valor de  $p$  fue  $< 0,05$ , lo que indica que las diferencias obtenidas son estadísticamente significativas.

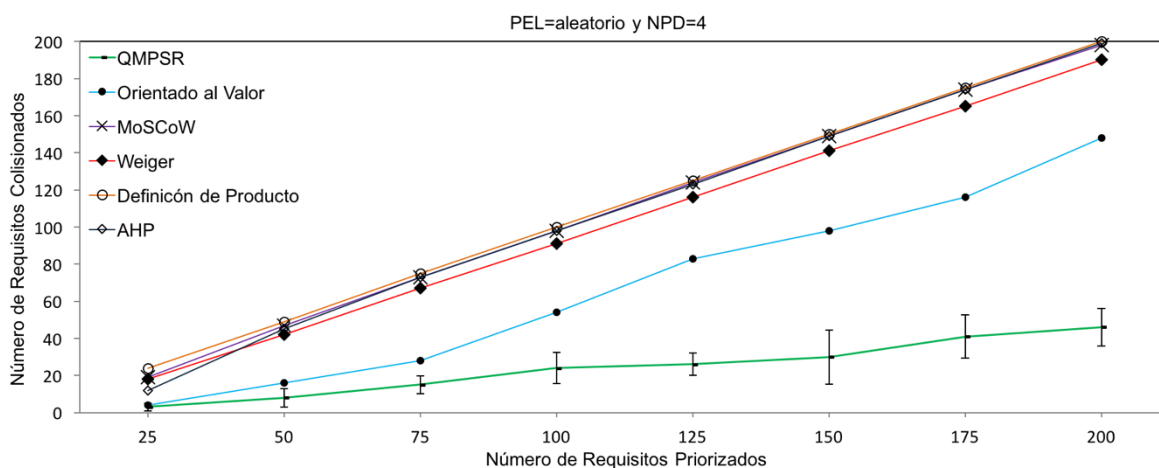
### 5.1.5. Discusión

En esta sección se analizan y discuten los resultados de los nueve experimentos, con el fin de encontrar respuestas a las preguntas de investigación **PI<sub>15</sub>**, **PI<sub>16</sub>** y **PI<sub>17</sub>**.

<sup>24</sup>  $SD$ =Desviación Estándar.

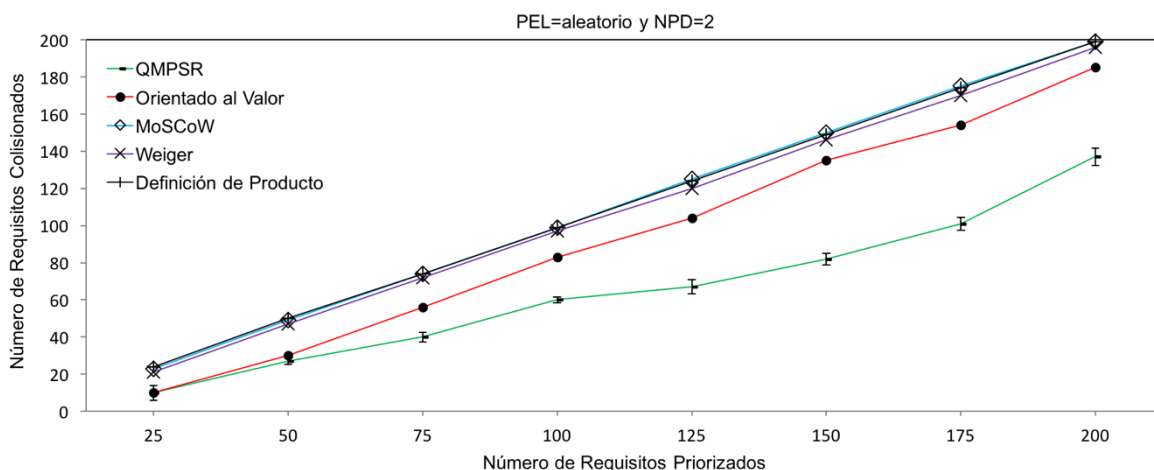
Los experimentos A, B y C proporcionan evidencia de las colisiones, generadas por cada método de priorización con un *PEL aleatorio* y un *NPD* de 4, 2 y 8, respectivamente. En todos los casos, QMPSR supera a todos los métodos comparados, ya que genera menos requisitos colisionados.

Por un lado, los resultados obtenidos a través del experimento A (ver Figura 69) muestran que QMPSR genera menos requisitos colisionados para todos los conjuntos de requisitos de entrada. Aun cuando el método Orientado al Valor obtiene el segundo mejor rendimiento (es decir, los métodos QMPSR y Orientado al Valor obtienen un promedio de 19,9% (SD=3,82) y 51.41% (SD=19,31) de requisitos colisionados, respectivamente), la diferencia con respecto a QMPSR aumenta cuando el número de requisitos de entrada crece.



**Figura 69:** Experimento A. Número de requisitos colisionados (eje y) para todos los métodos evaluados: QMPSR, Orientado al Valor, MoSCoW, Weiger, Definición de Producto y AHP. Valores obtenidos como promedio de 10 ejecuciones para el 25, 50, 75, 100, 125, 150, 175 y 200 requisitos priorizados (eje x), teniendo en cuenta *PEL=aleatorio* y *NPD=4*. Las barras de error (SD) se incluyen para mostrar la dispersión del método QMPSR.

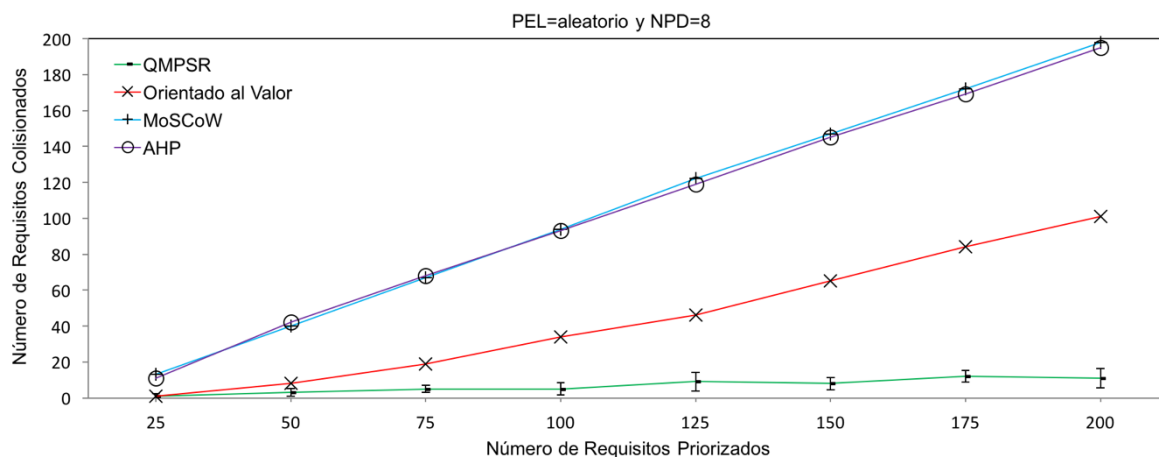
Por otra parte, el experimento B (ver Figura 70) también proporciona evidencia de menos requisitos colisionados para todos los conjuntos de requisitos. Sólo para los conjuntos de requisitos de 25 y 50, el método Orientado al Valor obtiene resultados similares a QMPSR. Aquí la diferencia de los requisitos colisionados entre el método Orientado al Valor y QMPSR es inferior al 12%, mientras que para todos los otros conjuntos de requisitos el número de requisitos colisionados es superior al 35%.



**Figura 70:** Experimento B. Número de requisitos colisionados (eje y) para todos los métodos evaluados: QMPSR, Orientado al Valor, MoSCoW, Weiger y Definición de Producto. Valores obtenidos como promedio de 10 ejecuciones para el 25, 50, 75, 100, 125, 150, 175 y 200 requisitos priorizados (eje x), teniendo en cuenta  $PEL=aleatorio$  y  $NPD=2$ . Las barras de error (SD) se incluyen para mostrar la dispersión del método QMPSR.

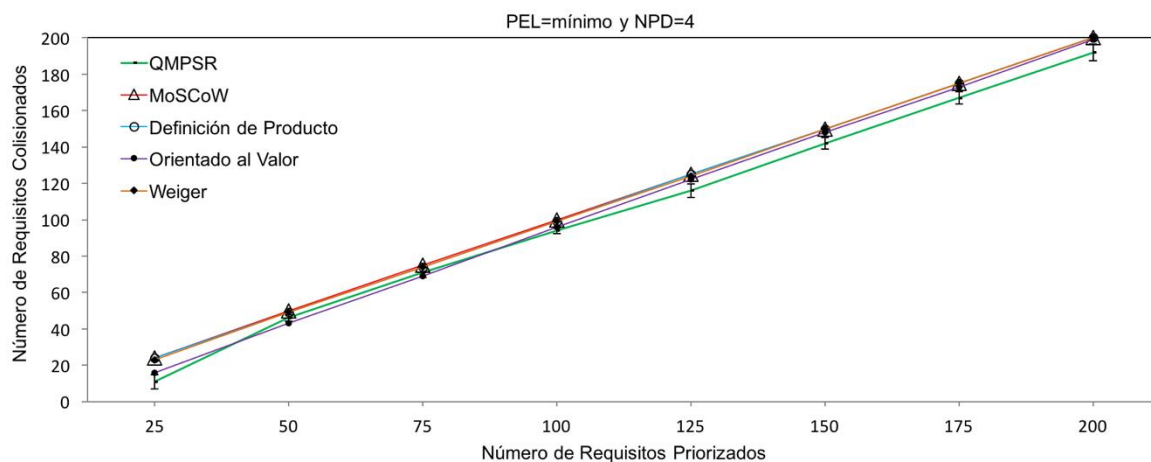
Además, es importante mencionar que en el experimento B, los métodos Orientado al Valor y QMPSR generan un mayor número de requisitos colisionados que en el experimento A: en el experimento B, dichos métodos producen un promedio de 64,00% (SD=41,88) y 182,83% (SD=33,83) de requisitos colisionados más altos que en el experimento A. Sin embargo, el experimento B implica un menor número de evaluaciones para cada conjunto de requisitos de entrada que el experimento A. Es decir, el experimento B presenta un promedio de 40,41% (SD=1,34) menos de esfuerzo de priorización que el experimento A (ver las dos primeras filas de la Tabla 28). Por lo tanto, estos métodos se ven afectados por una reducción del  $NPD$  y el esfuerzo de priorización (número de evaluaciones para cada conjunto de requisitos de entrada), especialmente QMPSR. Sin embargo, QMPSR genera menos requisitos colisionados en todos los conjuntos de requisitos, en comparación con todos los otros métodos.

Con respecto al experimento C (ver Figura 71), QMPSR genera menos requisitos colisionados que en los experimentos A y B. Por ejemplo, en el experimento C, QMPSR genera un promedio de 5,81% (SD=1,07) de requisitos colisionados, mientras que en los experimentos A y B se obtiene un 19,9% (SD=3,82) y 55,22% (SD=7,97), respectivamente. También vale la pena señalar que en el experimento C, el promedio de esfuerzo de priorización es 77,57% (SD=4,85) y 198,05% (SD=5,49) mayor que en los experimentos A y B, respectivamente (ver las tres primeras filas de la Tabla 28). Por lo tanto, se puede afirmar que cuando QMPSR se evalúa con un  $PEL$  aleatorio, se generan menos requisitos colisionados, mientras  $NPD$  y el esfuerzo de priorización aumentan. Del mismo modo, también se puede ver que las diferencias entre QMPSR y el resto de métodos aumentan cuando se consideran grandes conjuntos de requisitos.



**Figura 71:** Experimento C. Número de requisitos colisionados (eje y) para todos los métodos evaluados: QMPSR, MoSCoW, Orientado al Valor y AHP. Valores obtenidos como promedio de 10 ejecuciones para el 25, 50, 75, 100, 125, 150, 175 y 200 requisitos priorizados (eje x), teniendo en cuenta  $PEL=aleatorio$  y  $NPD=8$ . Las barras de error (SD) se incluyen para mostrar la dispersión del método QMPSR.

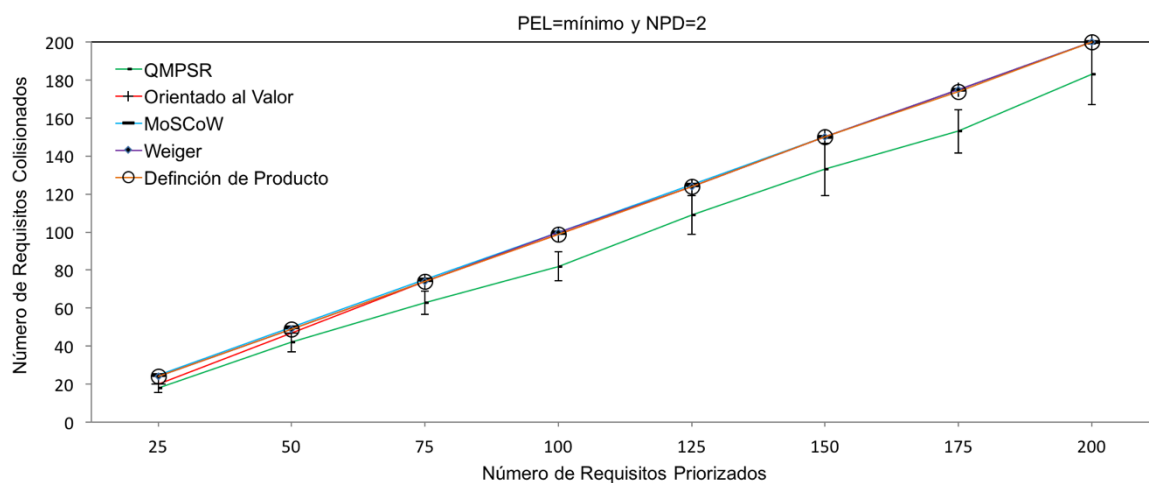
En cuanto a los experimentos D, E y F (ver la Figura 72, la Figura 73 y la Figura 74, respectivamente), éstos permiten analizar las colisiones generadas en los métodos de priorización cuando hay un  $PEL$  mínimo y un  $NPD$  de 4, 2 y 8, respectivamente. Los resultados obtenidos en estos experimentos proporcionan evidencia empírica de que todos los métodos comparados generan un número similar de requisitos colisionados para cada conjunto de requisitos.



**Figura 72:** Experimento D. Número de requisitos colisionados (eje y) para todos los métodos evaluados: QMPSR, MoSCoW, Orientado al Valor, Weiger y Definición de Producto. Valores obtenidos como promedio de 10 ejecuciones para el 25, 50, 75, 100, 125, 150, 175 y 200 requisitos priorizados (eje x), teniendo en cuenta  $PEL=mínimo$  y  $NPD=4$ . Las barras de error (SD) se incluyen para mostrar la dispersión del método QMPSR.

En el experimento D (ver Figura 72), QMPSR obtiene un promedio de 87,94% (SD=17,8) de requisitos colisionados, mientras que los otros métodos obtienen valores que alcanzan

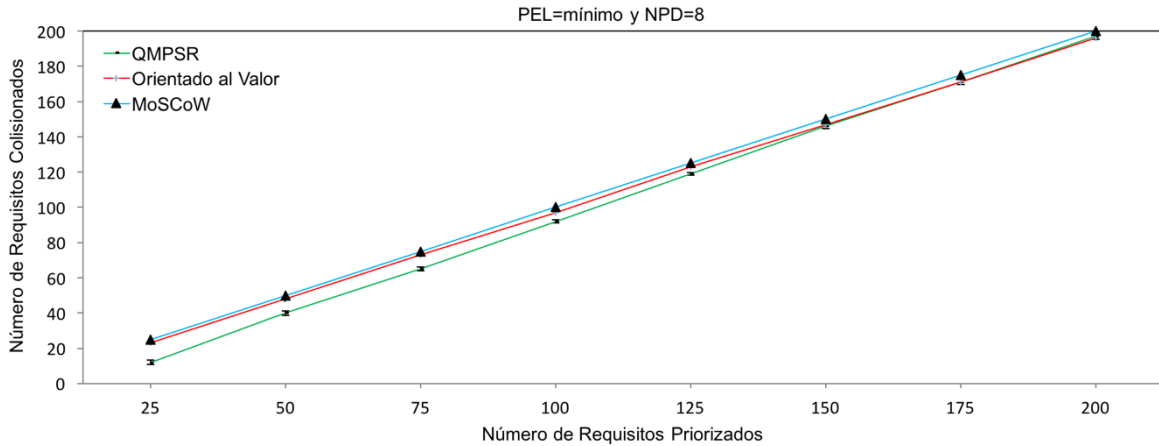
desde el 91%. Del mismo modo, en el experimento E (ver la Figura 73), QMPSR produce un promedio de 84,59% (SD=5,91) de requisitos colisionados, mientras que los valores obtenidos por los otros métodos alcanzan desde el 96%. Además, como se puede ver en la Figura 73, QMPSR genera menos requisitos colisionados en todos los conjuntos de requisitos. Por último, en el experimento F (ver la Figura 74), QMPSR genera un promedio de 86,92% (SD=16,97) de requisitos colisionados, mientras que los otros métodos obtienen los valores desde 96%. En este experimento, QMPSR destaca por encima de los otros para el conjunto de requisitos de 25, produciendo 48% de requisitos colisionados, mientras que otros métodos obtienen los valores desde 92%.



**Figura 73:** Experimento E. Número de requisitos colisionados (eje y) para todos los métodos evaluados: QMPSR, Orientado al Valor, MoSCoW, Weiger y Definición de Producto. Valores obtenidos como promedio de 10 ejecuciones para el 25, 50, 75, 100, 125, 150, 175 y 200 requisitos priorizados (eje x), teniendo en cuenta  $PEL=mínimo$  y  $NPD=2$ . Las barras de error (SD) se incluyen para mostrar la dispersión del método QMPSR.

Uno de los factores que puede explicar los resultados descritos anteriormente es la disminución considerable del esfuerzo de priorización en los experimentos D, E y F. Aunque los experimentos A, B y C incluyen el mismo  $NPD$  como los experimentos D, E y F, respectivamente, éstos últimos contienen sólo una de las dimensiones de priorización definidas (identificada aleatoriamente), para determinar el valor para cada requisito, es decir, generan un menor esfuerzo de priorización. Por ejemplo, en el experimento D, los métodos generan un promedio de 151,98% (SD=4,99) menos de esfuerzo de priorización que en el experimento A. Del mismo modo, en los experimentos E y F, los métodos generan una media de 50,08% (SD=1,67) y 347,26% (SD=5,54) menos de esfuerzo de priorización que en los experimentos B y D, respectivamente.

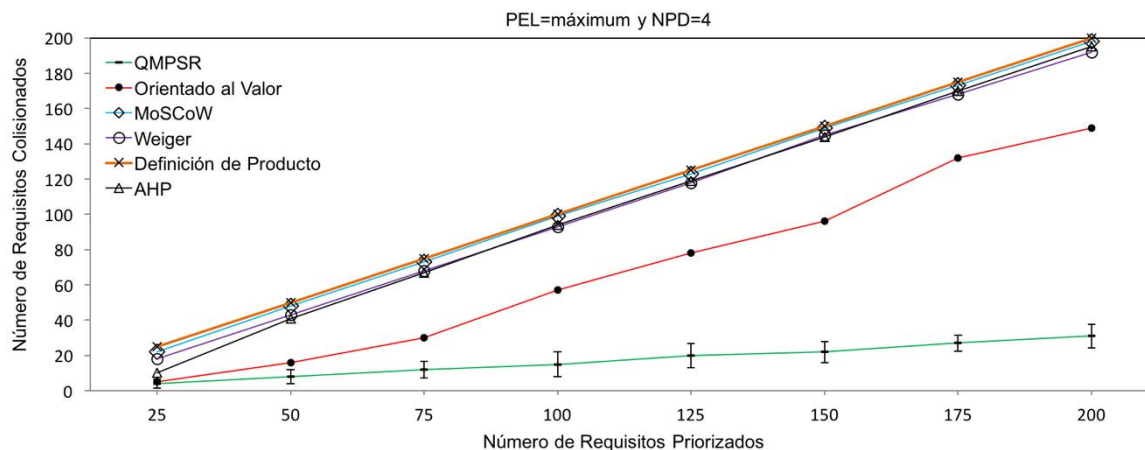




**Figura 74:** Experimento F. Número de requisitos colisionados (eje y) para todos los métodos evaluados: QMPSR, Orientado al Valor y MoSCoW. Valores obtenidos como promedio de 10 ejecuciones para el 25, 50, 75, 100, 125, 150, 175 y 200 requisitos priorizados (eje x), teniendo en cuenta *PEL*=mínimo y *NPD*=8. Las barras de error (SD) se incluyen para mostrar la dispersión del método QMPSR.

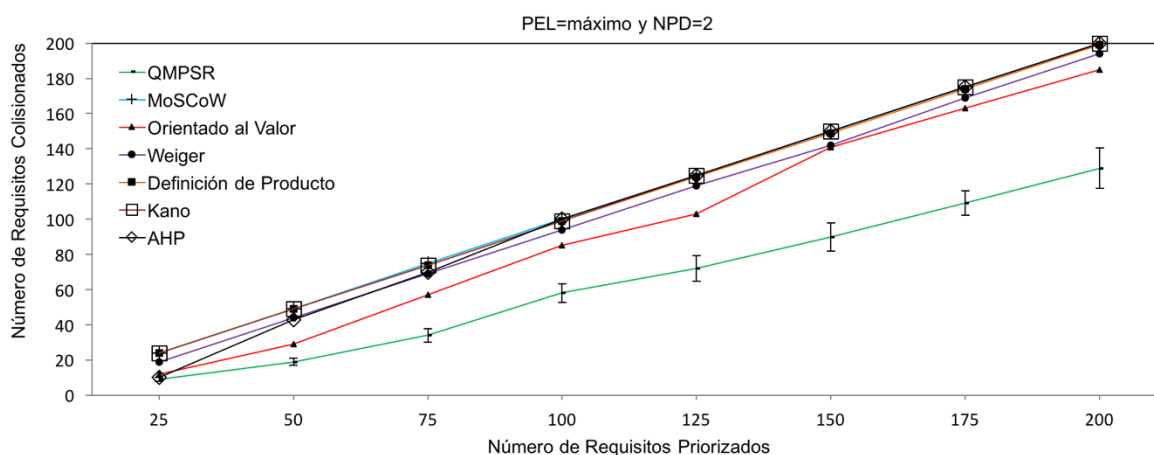
En cuanto a los experimentos G, H e I (ver la Figura 75, la Figura 76 y la Figura 77, respectivamente), éstos proporcionan las colisiones producidas por cada método de priorización con un *PEL* máximo y un *NPD* de 4, 2 y 8, respectivamente. Los resultados obtenidos en estos experimentos demuestran claramente que QMPSR supera a los otros métodos comparados.

Los resultados del experimento G (ver la Figura 75) muestran que QMPSR genera menos requisitos colisionados para todos los conjuntos de requisitos. Por otra parte, el QMPSR proporciona menos requisitos colisionados en comparación con otros experimentos que incluyen el mismo *NPD* (4), pero diferente *PEL* (*aleatorio* y *mínimo* en los experimentos A y D, respectivamente). Por ejemplo, en el experimento G, QMPSR genera un promedio de 4,33% y 36,63% menos de requisitos colisionados que en los experimentos A y D, respectivamente. Sin embargo, el experimento G implica un mayor número de dimensiones de priorización seleccionadas para determinar el valor de cada requisito que los experimentos A y D. Es decir, el experimento G genera un promedio de 58,79% (SD=3,11) y 300% (SD=0) de esfuerzo de priorización mayor que los experimentos A y D, respectivamente (ver la primera, cuarta y séptima filas de la Tabla 28). Por lo tanto, se puede afirmar que cuando QMPSR se evalúa con 4 dimensiones de priorización, teniendo en cuenta diferentes valores de *PEL*, hay menos requisitos colisionados, a medida que aumenta el esfuerzo de priorización.



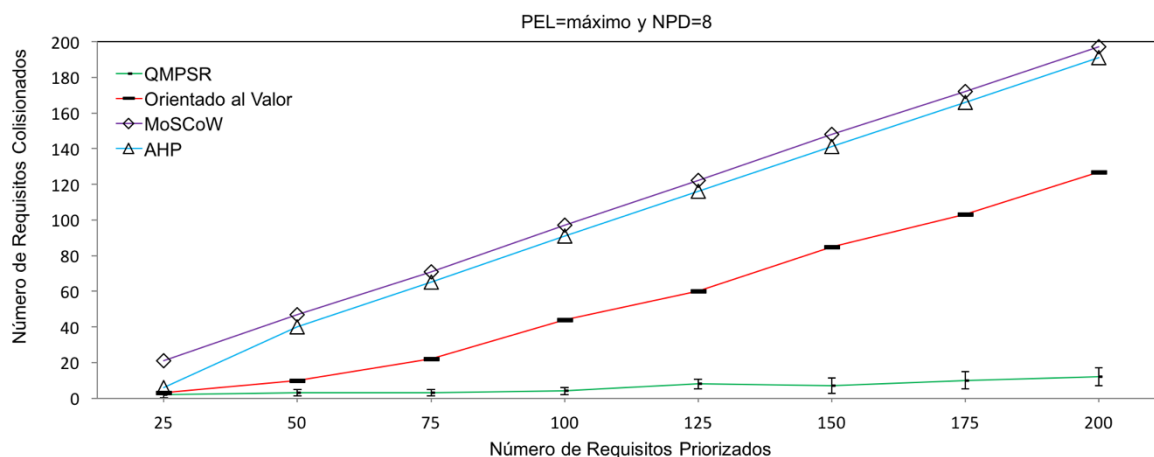
**Figura 75:** Experimento G. Número de requisitos colisionados (eje y) para todos los métodos evaluados: QMPSR, Orientado al Valor, MoSCoW, Weiger, Definición de Producto y AHP. Valores obtenidos como promedio de 10 ejecuciones para el 25, 50, 75, 100, 125, 150, 175 y 200 requisitos priorizados (eje x), teniendo en cuenta  $PEL=máximo$  y  $NPD=4$ . Las barras de error (SD) se incluyen para mostrar la dispersión del método QMPSR.

Del mismo modo, los resultados del experimento H (ver la Figura 76) ilustran que los métodos QMPSR, Orientado al Valor y AHP obtienen un número similar de requisitos colisionados para el conjunto de 25 requisitos priorizados. Sin embargo, la diferencia en el número de requisitos colisionados entre QMPSR y los otros métodos se hace más explícita a medida que el número de requisitos de entrada aumenta. En este experimento, QMPSR produce menos requisitos colisionados para todos los conjuntos de requisitos, en particular para los grupos de 25 y 50, donde el promedio de los requisitos colisionados es de sólo 37% (SD=1,41).



**Figura 76:** Experimento H. Número de requisitos colisionados (eje y) para todos los métodos evaluados: QMPSR, Orientado al Valor, MoSCoW, Weiger, Definición de Producto, Kano y AHP. Valores obtenidos como promedio de 10 ejecuciones para el 25, 50, 75, 100, 125, 150, 175 y 200 requisitos priorizados (eje x), teniendo en cuenta  $PEL=máximo$  y  $NPD=2$ . Las barras de error (SD) se incluyen para mostrar la dispersión del método QMPSR.

Los resultados obtenidos a partir del último experimento (ver el experimento I en la Figura 77) demuestran que QMPSR también genera menos requisitos colisionados para todos los conjuntos de requisitos. En este experimento, QMPSR sólo produce un promedio de 5,59% (SD=1,34) de requisitos colisionados, mientras que los otros métodos obtienen valores que alcanzan desde 63%. El método Orientado al Valor aparece en la segunda posición, pero para los conjuntos de requisitos de 25, 50 y 75 muestra un promedio de 20,44% (SD=8,67) de requisitos colisionados, mientras que QMPSR obtiene un promedio de 6% (SD=2) para el mismo conjunto de requisitos de entrada.



**Figura 77:** Experimento I. Número de requisitos colisionados (eje y) para todos los métodos evaluados: QMPSR, Orientado al Valor, MoSCoW y AHP. Valores obtenidos como promedio de 10 ejecuciones para el 25, 50, 75, 100, 125, 150, 175 y 200 requisitos priorizados (eje x), teniendo en cuenta *PEL*=máximo y *NPD*=8. Las barras de error (SD) se incluyen para mostrar la dispersión del método QMPSR.

Se puede observar que en los experimentos G, H e I (*PEL*=máximo), los métodos evaluados utilizan todas las dimensiones de priorización definidas para determinar el valor para cada requisito. Por lo tanto, el experimento I produce un promedio de 200% (SD=0) y 400% (SD=5,49) de esfuerzo de priorización mayor que los experimentos G y H, respectivamente (ver las tres últimas filas de la Tabla 28). En general, QMPSR genera menos requisitos colisionados en el experimento I que en los experimentos G y H. Por ejemplo, en el experimento I, QMPSR produce un promedio de 191,78% (SD=67,12) y 902,17% (SD=330,05) menos de requisitos colisionados que en los experimentos G y H, respectivamente. Por lo tanto, se puede afirmar que cuando QMPSR tiene un *PEL* máximo, hay menos requisitos colisionados, a medida que *NPD* aumenta.

Es importante señalar que QMPSR obtiene los mejores resultados en todos los experimentos que implican un *PEL* aleatorio o máximo y 8 dimensiones de priorización. De hecho, la media de los requisitos colisionados es menor a 6% en estos casos. Del mismo modo, en los experimentos con 4 dimensiones de priorización y un *PEL* aleatorio o máximo, la media de los requisitos colisionados es menor a 20%. Por el contrario, QMPSR genera un mayor número de requisitos colisionados en todos los experimentos con un *PEL*

*mínimo*, obteniendo un promedio de alrededor del 86% de requisitos colisionados, mientras que los otros métodos obtienen valores que alcanzan desde el 91%.

Así pues, los resultados obtenidos proporcionan evidencia suficiente para responder las preguntas de la investigación **PI<sub>15</sub>**, **PI<sub>16</sub>** y **PI<sub>17</sub>**, indicadas anteriormente en la sección 1.2:

- **PI<sub>15</sub>**: ¿Es posible reducir las colisiones de requisitos a través del método de priorización propuesto (QMPSR) dirigido por elementos cualitativos?

Los resultados obtenidos a partir de todos los experimentos demuestran que QMPSR reduce el número de requisitos colisionados, en comparación con todos los otros métodos. El método de priorización propuesto se basa en relaciones que se establecen entre los elementos de los aspectos y los requisitos, permitiendo diferenciar mejor entre los requisitos asociados a elementos con la misma prioridad, gracias al Factor de Asociación previamente definido, produciendo resultados precisos en la clasificación final de requisitos.

Mediciones	QMPSR	Orientado al Valor	MoSCoW	Weiger	Definición de Producto	AHP	Kano
Media	46,03%	68,68%	97,03%	93,93%	99,08%	87,42%	98,95%
SD	35,19	24,6	3,68	4,15	0,49	3,45	0
Máx.	87,94%	96,8%	100%	98,98%	100%	91,24%	98,95%
Mín.	5,59%	32,24%	88,52%	89,05%	98,64%	82,35%	98,95%

**Tabla 29:** Porcentaje de requisitos colisionados generados por los métodos a través de todos los experimentos.

- **PI<sub>16</sub>**: ¿En qué medida QMPSR supera a otros métodos de priorización en términos de colisiones de requisitos y de esfuerzo de priorización?

Para demostrar cómo QMPSR supera al resto de métodos en términos de requisitos colisionados, en la Tabla 29, se resumen los porcentajes de requisitos colisionados, producidos por los métodos a través de todos los experimentos. En general, manteniendo el mismo esfuerzo de priorización para todos los métodos en cada experimento, QMPSR produce menos requisitos colisionados. Los resultados obtenidos muestran que QMPSR supera a todos los métodos comparados, obteniendo una media de 46,03% (SD=35,19) de requisitos colisionados. Este mejor resultado es seguido por los métodos Orientado al Valor, AHP, Weiger, MoSCoW, Kano y Definición de Producto, que producen los siguientes promedios de requisitos colisionados: 68,68% (SD=24,6), 87,42% (SD=3,45), 93,93% (SD=4,15), 97,03% (SD=3,68), 98,95% (SD=0) y 99,08% (SD=0,49), respectivamente. Vale la

pena mencionar que el valor mínimo obtenido por QMPSR es en el experimento I (5,59% de requisitos colisionados), donde, como se ha analizado anteriormente, QMPSR obtiene los mejores resultados en condiciones adversas.

Mediciones	QMPSR	Orientado al Valor	MoSCoW	Weiger	Definición de Producto	AHP	Kano
Media	0,49	0,78	0,99	0,96	0,99	0,96	0,99
SD	0,38	0,2	0,01	0,02	0,01	0,02	0
Máx.	0,96	0,99	1	0,99	1	0,99	0,99
Mín.	0,05	0,43	0,97	0,93	0,99	0,93	0,99

**Tabla 30:** Pendientes de las rectas ajustadas para los métodos comparados en todos los experimentos.

- **PI17:** ¿Cómo se comporta el método QMPSR en términos de escalabilidad, en función del número de colisiones de requisitos observadas, cuando el conjunto de requisitos aumenta en tamaño?

Para mostrar el comportamiento del QMPSR en términos de escalabilidad cuando el número de requisitos de entrada crece, se analizaron los gráficos obtenidos a través de todos los experimentos (desde la Figura 69 hasta la Figura 77). De esta manera, se calculó una recta de ajuste con los valores promedio en todos los experimentos para todos los métodos comparados, obteniendo un valor  $R^2$  medio satisfactorio ( $M=0,99$ ,  $SD=0,01$ ), lo que permite obtener una tendencia aceptable considerando la recta de ajuste como un buen elemento predictivo del comportamiento de cada método en términos de escalabilidad. Por lo tanto, la escalabilidad se estudió en términos de la pendiente de las rectas resultantes de todos los métodos, en cada experimento, lo que indica cómo cada método se comporta cuando aumenta el número de requisitos de entrada. De esta manera, un valor de pendiente más cercano a 1 indica un crecimiento lineal de las colisiones, mientras el número de requisitos de entrada aumenta. Por otro lado, un valor de la pendiente más cercano a 0 corresponde al caso ideal para un buen comportamiento en términos de escalabilidad, ya que la pendiente de la recta (primera derivada) es casi constante. Por el contrario, un valor de pendiente de más de 1 indica un crecimiento rápido de los requisitos colisionados, mientras el número de requisitos de entrada aumenta (la peor situación). La Tabla 30 informa de las pendientes de las rectas ajustadas para cada método a través de los experimentos. Como se puede ver, QMPSR supera a todos los métodos comparados, obteniendo una pendiente de 0,49 ( $SD=0,38$ ),

siendo una decantación por debajo de 1 y no muy lejos de 0. Este mejor resultado es seguido por métodos Orientado al Valor, Weiger, AHP, Definición de Producto, MoSCoW y Kano, que producen los siguientes promedios: 0,78 (SD=0,2), 0,96 (SD=0,02), 0,96 (SD=0,02), 0,99 (SD=0,01), 0,99 (SD=0,01) y 0,99 (SD=0), respectivamente. También vale la pena mencionar el valor de la pendiente mínima obtenida por QMPSR en el experimento I (0,05). En este experimento, la pendiente de la recta se vuelve casi constante, siendo un buen indicador de cómo QMPSR se comporta en condiciones adversas.

### **5.1.6. Recapitulación**

A lo largo de esta sección, se ha presentado una descripción completa del marco experimental, propuesto para evaluar la capacidad del método de priorización QMPSR, para gestionar ágilmente los requisitos en Scrum-UIA, incluyendo los resultados y discusiones más importantes. El marco comparativo propuesto alcanza un alto grado de normalización, homogeneidad y reutilización de diferentes conceptos y componentes, que permiten llevar a cabo las comparaciones y evaluaciones de las mismas características, en diferentes métodos de priorización.

Utilizando este marco experimental, se han definido las métricas de esfuerzo de priorización y de colisión de requisitos, para llevar a cabo nueve experimentos con diferentes conjuntos de requisitos de entrada, comparando QMPSR con seis conocidos métodos de priorización existentes (MoSCoW, Orientado al Valor, Weiger, Definición de Producto, AHP y Kano). Estos experimentos permiten llevar a cabo un análisis en profundidad de las colisiones generadas en los métodos de priorización seleccionados.

Los resultados de los experimentos proveen una respuesta a las preguntas de investigación establecidas. De esta manera, es posible afirmar que, manteniendo el mismo esfuerzo de priorización para todos los métodos en cada experimento, QMPSR proporciona mejores resultados, superando uniformemente a todos los métodos de priorización con un PEL (Nivel de Esfuerzo de Priorización) aleatorio o máximo, sin tener en cuenta el NDP (Número de Dimensiones de Priorización). Además, se generan menos requisitos colisionados a medida que aumentan el NPD y los requisitos de entrada. Esto pone de relieve la capacidad de QMPSR para adaptarse a entornos complejos y dinámicos, presentando un buen comportamiento en términos de escalabilidad.

Finalmente, los resultados experimentales obtenidos permiten corroborar la hipótesis de partida **H2.2**, al concluir que QMPSR permite conducir, de manera dinámica, el proceso de priorización de los requisitos en Scrum-UIA, generando un ranking final que posibilita disminuir las colisiones de requisitos y presentar un buen comportamiento en términos de escalabilidad.

## 5.2. Evaluación de la Usabilidad de InterArch

Una vez presentada la experimentación para evaluar el método de priorización de Scrum-UIA (QMPSR) frente a otros métodos, en esta sección se describe la evaluación llevada a cabo para tener algunas pistas sobre la usabilidad de la herramienta InterArch. Esta herramienta sistematiza la técnica (InterArch-T), que se propone para promover el desarrollo incremental en Scrum-UIA a través de la AI. Para la validación de la usabilidad de dicha herramienta, se ha realizado un diseño cuasi experimental con 12 Arquitectos de la Información. En primer lugar, en la sección 5.2.1, se describen las variables, las preguntas de investigación, los participantes y las tareas de la experimentación. Posteriormente, en la sección 5.3.2, se detallan los pasos realizados en la ejecución del experimento. Seguidamente, en la sección 5.2.3, se describen y analizan los diferentes resultados obtenidos en la sesión experimental con los 12 Arquitectos de la Información y en el cuestionario de usabilidad. Finalmente, en la sección 5.3.4, se realiza una discusión respecto a los resultados generales obtenidos en la experimentación.

Además, con la realización de esta evaluación, también se pretende corroborar la hipótesis de partida **H2.3**: Es posible promover el desarrollo incremental en la metodología anteriormente ideada (hipótesis de partida H2.1) a partir de las descripciones conceptuales que el Arquitecto de la Información crea, dentro del dominio del problema, y que evoluciona a fases más cercanas al dominio de la solución. Asimismo, es posible sistematizar esta técnica mediante la construcción de una herramienta CASE (Computer Aided Software Engineering), fácil de usar y de aprender por el Arquitecto de la Información.

### 5.2.1. Marco Experimental

El objetivo principal de la experimentación fue estudiar y analizar el grado de usabilidad de la herramienta InterArch. Para medir el grado de usabilidad lograda por esta herramienta, se ha procedido a diseñar un experimento sobre un conjunto de usuarios potenciales. De esta manera, se ha realizado un diseño cuasi experimental, debido a que el grupo de estudio no se ha seleccionado de manera aleatoria, sino del ambiente cercano al investigador.

Asimismo, el experimento se ha diseñado para cumplir con las condiciones de usabilidad, indicadas en el estándar ISO-9241-11 (DIN, 1998), que permite averiguar en qué medida un producto se puede utilizar por determinados usuarios para conseguir objetivos específicos con eficacia, eficiencia y satisfacción. De este modo, la aplicación del estándar anterior facilitaría conseguir diferentes métricas, con el fin de optimizar la herramienta en base a la experiencia de los usuarios.

Para efectuar la evaluación de la herramienta CASE InterArch, se han seleccionado la técnica *Test Retrospectivo* y el protocolo *Thinking Aloud*. La técnica *Test Retrospectivo* consiste en la revisión de los registros de videos que se almacenan durante la realización del test a los usuarios, permitiendo, de esta forma, recopilar información adicional (Nielsen,

1993). Una de las desventajas es el tiempo requerido, puesto que se debe revisar cada grabación de test por lo menos dos veces.

Por otro lado, el protocolo *Thinking Aloud*, introducido por Clayton Lewis (Lewis, 1982), es considerado uno de los más valiosos en ingeniería de la usabilidad (Holzinger, 2005). Se trata de tener al usuario final continuamente pensando en voz alta mientras se utiliza el sistema. Al verbalizar los pensamientos de los usuarios, la técnica permite entender cómo ellos ven el sistema computacional, lo que hace que sea más fácil identificar conceptos erróneos (Nielsen, 1993). Para la implementación de este protocolo, se ha solicitado a los usuarios participantes que indicaran en voz alta sus sentimientos y opiniones, mientras interactuaban con la herramienta InterArch en la realización de una tarea específica.

Tanto para el *Test Retrospectivo* como para el protocolo *Thinking Aloud*, se requería capturar la interacción con la herramienta InterArch en la pantalla, registrando el audio y el video de los usuarios durante toda la sesión experimental. En base a estas exigencias, se ha utilizado la aplicación Camtasia Studio (Camtasia-Studio, 2011), que permite esa funcionalidad exigida.

#### 5.2.1.1. Variables y Preguntas de Investigación

Teniendo en cuenta el objetivo principal de la experimentación, se han definido las variables y preguntas de la investigación. En cuanto a las variables independientes, éstas fueron: el número de tareas realizadas por los usuarios, el tiempo total utilizado por los usuarios para terminar las tareas, los rasgos de los participantes (edad y género), y las respuestas de los usuarios a las preguntas de un cuestionario de usabilidad.

Por otro lado, las variables dependientes son relativas a las tres condiciones de usabilidad, indicadas en el estándar ISO-9241-11: eficacia, eficiencia y satisfacción. De esta manera, la eficacia corresponde al grado de éxito que los usuarios alcanzan en la ejecución de las tareas, que es medida como el porcentaje de consecución de las tareas por los usuarios. La eficiencia consiste en el tiempo que los usuarios requieren para completar cada tarea, y se mide en segundos. Por último, la satisfacción es la percepción subjetiva de usabilidad que se alcanza con la herramienta InterArch, y que se mide a través del valor medio de las respuestas de los usuarios a las preguntas del cuestionario de usabilidad.

Asimismo, se han definido las preguntas de investigación para responder las tres variables dependientes, que corresponden a (presentadas en la sección 1.2):

**PI18:** ¿Cuál es la eficacia alcanzada por los usuarios en la realización de un modelo de contenido utilizando InterArch?

**PI19:** ¿Cuál es la eficiencia alcanzada por los usuarios en la realización de un modelo de contenido utilizando InterArch?

**PI20:** ¿Cuál es la percepción de usabilidad de los usuarios con respecto a la herramienta InterArch?



### **5.2.1.2. Participantes**

En la evaluación de la herramienta, participaron 12 personas que trabajan de forma habitual en empresas y consultoras de proyectos software, específicamente en tareas concernientes a la AI. Fueron 9 hombres y 3 mujeres, con edades comprendidas entre 24 y 43 años ( $M=32$ ,  $SD=8,06$ ). La alta dispersión en las edades de los usuarios permite indicar que la muestra de la población no estuvo sesgada a un estrato específico de edades, pudiendo considerarse esta diversidad en las edades de los usuarios como un aspecto positivo para la realización de la evaluación de la herramienta. En general, los usuarios contaban con conocimiento de partida sobre análisis y documentación en la estructuración y categorización de contenidos de sitios Web.

### **5.2.1.3. Tareas**

Para efectuar el experimento, se solicitó a cada usuario que se metiera en el papel del Arquitecto de la Información y realizara un modelo de contenido, relacionado con el proceso actual de creación de registros de anuncios de una empresa de productos de segunda mano. Tal proceso se efectúa en base a un registro por formulario impreso que permite obtener información sobre el detalle del anuncio y los datos particulares del anunciante. En concreto, se dispuso de una “maqueta” de fidelidad media, sacada de uno de los anuncios de la versión impresa de la revista Segunda Mano, y se solicitó crear el modelo de contenido utilizando la herramienta InterArch.

A continuación, se presenta el enunciado ficticio, utilizado para que los usuarios pudieran realizar el experimento:

*Una revista de anuncios de productos de segunda mano, que en la actualidad se distribuye a partir de su versión impresa, quiere cambiar a una versión digital. Inicialmente, la empresa pretende modernizar su proceso actual de creación de registros de anuncios. El registro de anuncios se efectúa en base a un registro por formulario impreso, que permite obtener información del detalle del anuncio y de los datos particulares del anunciante. Más específicamente, la empresa requiere que el actual registro de anuncios sea a través de una página Web, que incluya información sobre la provincia, categoría y tipo de anuncio, junto con el nombre, email y teléfono del anunciante, así como datos específicos del anuncio, referentes al título, descripción, precio (fijo, negociable o gratis), video e imagen del producto.*

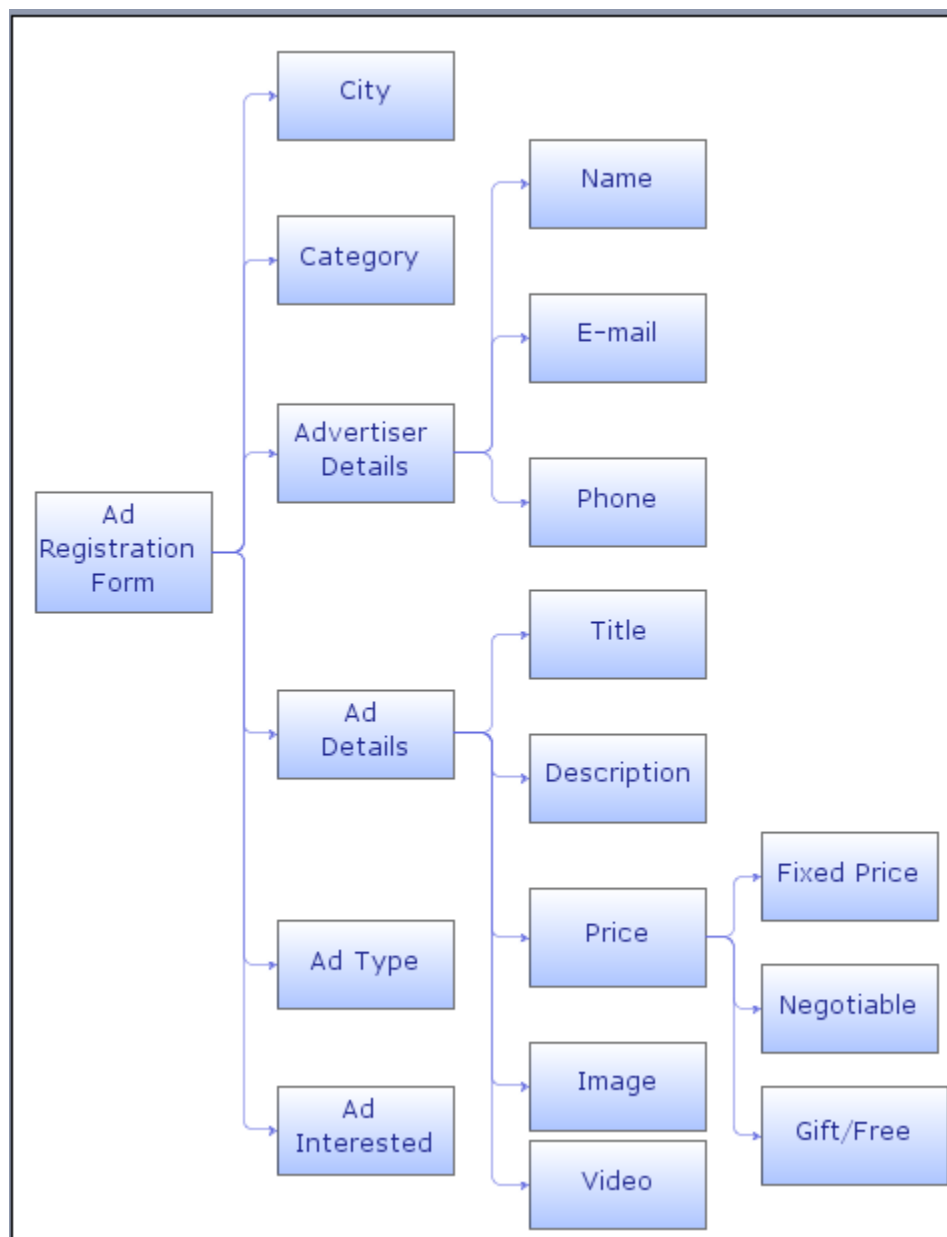
*El trabajo se ha encargado a la empresa informática donde usted trabaja como Arquitecto de la Información. A partir de la maqueta proporcionada, que está sacada de uno de los anuncios de la versión impresa de la revista de venta Segunda Mano (ver Figura 78), deberá trabajar en la parte de modelado conceptual, llevando a cabo las siguientes tareas:*

- *Segmentación del contenido del registro de anuncios.*
- *Creación de un modelo de contenido, utilizando la herramienta InterArch.*

Ad Registration Form	
City	Pulaski
Category	Sport
Ad Type	For Sale
Ad Interested	Particular
<b>Advertiser Details</b>	
Name	John Perez
E-mail	John-perez@email.com
Phone	85558548
<b>Ad Details</b>	
Title	Bicycle for children
Description	Perfect conditions: Damping front and center.
<b>Price</b>	
Fixed Price	90 US
Negotiable	No
Gift/Free	No
Image	
Video	

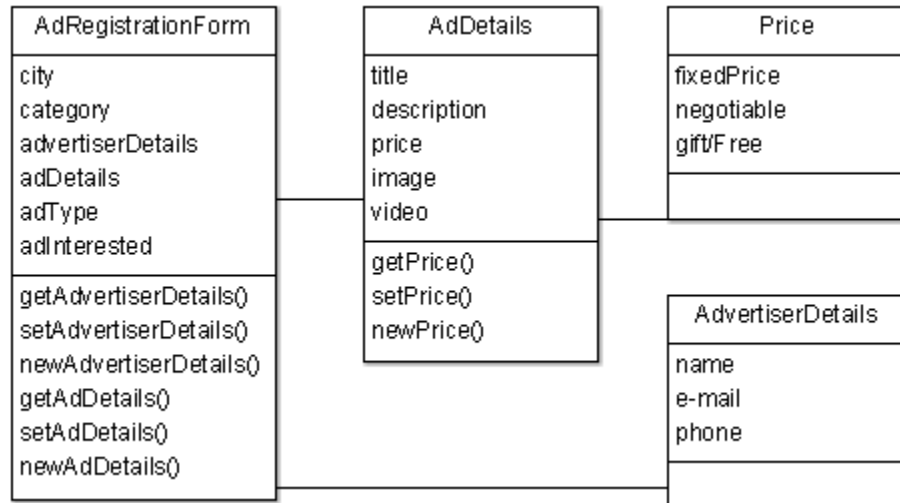
**Figura 78:** Maqueta de registro de anuncios para productos de segunda mano.

Utilizando la herramienta InterArch, se ha elaborado el posible modelo de contenidos, cuya composición se muestra en la Figura 79. Tal y como se puede observar, el modelo está compuesto principalmente por el elemento Ad Registration Form, que a la vez está compuesto por los elementos City, Category, Advertiser Details, Ad Details, Ad Type y Ad Interested. El elemento Advertiser Details se compone de los elementos Name, E-mail y Phone mientras que el elemento Ad Details está compuesto por los elementos Title, Description, Price, Image y Video. Finalmente, el elemento Price contiene elementos Fixed Price, Negotiable y Gift/Free.



**Figura 79:** Modelo de contenido de registro de anuncios para productos de segunda mano.

A continuación, en la Figura 80, se presenta el diagrama de clases UML como resultado al aplicar el conjunto de reglas de transformación de la herramienta InterArch al modelo de contenido de la Figura 79. Dicho diagrama está compuesto por las clases `AdRegistrationForm`, `AdDetails`, `Price` y `AdvertiserDetails`. Por su parte, cada clase está compuesta por atributos, operaciones y asociaciones, de acuerdo al conjunto de reglas de transformación, mencionadas en la sección 3.4.4. Por otro lado, las reglas de configuración que actúan por defecto generan clases, atributos y métodos con una visibilidad de tipo *public*. Además, las asociaciones entre clases se generan sin navegabilidad ni agregación entre ellas.



**Figura 80:** Diagrama de clases UML de registro de anuncios para productos de segunda mano.

### 5.2.2. Ejecución de la Experimentación

La experimentación sólo se ha realizado de forma presencial, en la cual se acordaba un lugar físico para efectuar la reunión. Durante la experimentación con los usuarios, se realizaron los siguientes pasos:

1. En primer lugar, para cada usuario individualmente, se realizó una demostración-tutorial de las distintas funcionalidades y características de la herramienta InterArch, que tuvo una duración promedio de 7 minutos y 48 segundos (SD=2 minutos y 22 segundos).
2. A continuación, como parte de la tarea a realizar en el experimento, se solicitó a los usuarios que se pusieran en el papel del Arquitecto de la Información y realizaran dos tareas (una segmentación de contenido y un modelado de contenido, utilizando la herramienta InterArch). Estas tareas se han presentado en detalle en la sección 5.2.1.3 del presente capítulo. Asimismo, se ha grabado a los usuarios en video y audio en cada sesión. Esto permitió contar con un registro tanto de las interacciones con la herramienta InterArch, como de las expresiones y comentarios efectuados durante la realización de la tarea.
3. En el último paso de la experimentación, y una vez finalizada la sesión con los usuarios, se les solicitó a cada uno de los participantes que rellenaran un cuestionario para evaluar la usabilidad de la herramienta InterArch. El cuestionario anterior se basaba en el cuestionario USE (Lund, 2001), con algunas modificaciones provistas por otros dos cuestionarios: Percepción de Utilidad y Facilidad de Uso de Davis (Davis, 1989) y Usabilidad de Purdue (Lin, 1997). El cuestionario que los participantes de la evaluación de la herramienta Scrum-UIA-MAT rellenaron se ha dividido en seis partes (ver Anexo A):

- La primera parte tiene preguntas para conseguir datos referentes al perfil del usuario.
- La segunda, tercera, cuarta y quinta partes tienen preguntas cerradas para medir las variables de *Utilidad*, *Facilidad de Uso*, *Facilidad de Aprendizaje* y *Satisfacción* de la herramienta InterArch, respectivamente. Las variables utilizadas anteriormente se corresponden con las cuatro dimensiones medidas para la estimación de la usabilidad.
- La última sección tiene preguntas abiertas, con el fin de conseguir información relacionada con los aspectos positivos y negativos de la herramienta InterArch.

### 5.2.3. Resultados Obtenidos

Esta sección describe y analiza los diferentes resultados, obtenidos en la sesión experimental con los 12 usuarios y en el cuestionario de usabilidad.

#### 5.2.3.1. Resultados de la Sesión Experimental

En esta sección, se detallan los resultados de los experimentos que se presentan en base al orden de actividades, establecidas para el proceso de evaluación de la herramienta InterArch.

##### *a) Demostración-tutorial de las funcionalidades y características de la herramienta InterArch*

La evaluación de la herramienta InterArch comenzó con una introducción al usuario, a modo de tutorial, presentando las principales funcionalidades y características de la herramienta. La introducción duró en promedio 7 minutos y 48 segundos (SD=2 minutos y 22 segundos). 14 minutos y 3 segundos fue el tiempo máximo que uno de los usuarios ha requerido, mientras que el usuario más rápido necesitó 5 minutos y 23 segundos. Teniendo en cuenta que había un total de 12 usuarios, la duración total ha sido de 1 hora, 33 minutos y 41 segundos. La dispersión de los tiempos empleados en esta tarea refleja el nivel de interacción con los usuarios, ya que en algunos casos se produjeron preguntas respecto a ciertas características de la herramienta. En otros casos, y en función del grado de asimilación personal, algunos usuarios apenas hicieron preguntas.

Esta introducción se ha realizado para familiarizar a los usuarios con las características básicas de la herramienta InterArch, para que conozcan las capacidades que ésta ofrece en términos generales. La herramienta InterArch no causó mayores dudas respecto a sus funcionalidades y características, puesto que replicaba muchas de las características de herramientas similares que los Arquitectos de la Información suelen emplear. Algunas de estas características corresponden a:

- La estructura de diseño de la herramienta InterArch cuenta con una paleta de elementos visuales, un entorno de trabajo y opciones de formato.
- La manipulación de los elementos se inicia al arrastrar los elementos al entorno de trabajo desde la paleta de elementos visuales.
- Los elementos de contenido se relacionan mediante el uso de los elementos de enlace o de forma directa, en base a seleccionar y arrastrar desde el elemento de contenido de origen hasta el elemento de contenido de destino.
- Los elementos del diagrama se pueden agrupar y manipular en bloque.
- Los elementos se pueden crear mediante mecanismos de herencia, seleccionando y arrastrando los elementos creados en el entorno de trabajo.
- El control de errores se consigue a base de deshacer y rehacer acciones. Es decir, los usuarios pueden volver a estados anteriores y rehacer acciones a través de estas funcionalidades. Esto se puede conseguir a través de los iconos de la barra de herramientas o directamente con la combinación de las teclas Ctrl+Z (deshacer) y Ctrl+Y (rehacer).
- La introducción de información en los elementos de contenido, enlace y etiquetado se realiza al hacer doble clic sobre dichos elementos.

*b) Segmentación de contenido*

Una vez realizada la introducción sobre las características de la herramienta InterArch, los usuarios recibieron un enunciado con la descripción de la tarea. El enunciado completo se encuentra en la sección 5.2.1.3, y la primera tarea consistió en la realización de una segmentación del contenido, utilizando la información proporcionada.

El primer punto medido ha sido la eficacia obtenida en la realización de esta tarea. Ésta se ha calculado a través del grado de éxito en la finalización de la tarea por parte de los usuarios. Los resultados muestran que la tarea de segmentación de contenido no exigió un mayor esfuerzo por parte de los usuarios, efectuándose sin mayores inconvenientes y de forma exitosa en todos los casos. Por lo tanto, se puede indicar que se ha alcanzado un 100% de eficacia.

El siguiente punto medido en esta tarea ha sido la eficiencia que se obtuvo en su realización, considerando el tiempo que los usuarios requirieron para llevarla a cabo. De este modo, la segmentación del contenido tuvo una duración promedio de 7 minutos y 5 segundos (SD=3 minutos y 12 segundos). 12 minutos y 24 segundos fue el tiempo máximo que uno de los usuarios ha requerido, mientras que 2 minutos y 55 segundos fue el tiempo mínimo que se requirió. La notable diferencia entre el tiempo máximo y el mínimo se debe, principalmente, al hecho de que dos usuarios

decidieron crear directamente el modelado de contenido utilizando la herramienta InterArch, dedicando, por lo tanto, menos tiempo al preámbulo (2 minutos y 55 segundos, y 3 minutos y 56 segundos). No obstante, había tres usuarios que dedicaron más de 10 minutos para llevar a cabo la tarea de segmentación del contenido (12 minutos y 24 segundos, 11 minutos y 50 segundos, y 10 minutos y 9 segundos). Sin embargo, no se presenta una alta dispersión en los tiempos empleados por los usuarios con respecto al tiempo promedio (7 minutos y 5 segundos) cuando se trata de la diferencia entre el tiempo máximo y el mínimo. Aquello se puede apreciar en el valor de la desviación estándar de 3 minutos y 12 segundos. Esta etapa tuvo una duración total de 1 hora, 25 minutos y 2 segundos con los 12 usuarios.

*a) Modelado de contenido utilizando la herramienta InterArch*

Por último, se les pidió a los usuarios que elaboraran un modelo de contenido, utilizando la herramienta InterArch, a partir de la segmentación de contenido efectuada anteriormente. Los usuarios han sido capaces de manipular, sin mayor dificultad ni esfuerzo, las distintas funcionalidades y características de la herramienta InterArch. Aunque el objetivo de tal tarea no era medir la validez de los modelos de contenido que los usuarios crearon, es importante indicar que la gran mayoría de ellos corresponde al que se esperaba en base a la tarea planteada, pudiendo observarse en la Figura 79. Aquello demuestra la experiencia de los usuarios en el área de análisis y categorización de contenidos de sitios Web.

El primer punto medido en esta tarea ha sido la eficacia obtenida en su realización. Los resultados muestran que en todos los casos los usuarios realizaron el modelado de contenido utilizando la herramienta InterArch correctamente. Por lo tanto, se puede señalar que, al igual que en la tarea anterior, se ha conseguido un 100% de eficacia.

El siguiente punto que se ha medido en esta tarea ha correspondido a la eficiencia que se obtuvo en su realización. Así pues, la elaboración del modelo de contenido, utilizando la herramienta InterArch, tuvo una duración promedio de 11 minutos y 54 segundos ( $SD=3$  minutos y 14 segundos). El tiempo máximo que uno de los usuarios requirió fueron 18 minutos y 45 segundos, mientras que el tiempo mínimo fueron 8 minutos y 40 segundos. A pesar de que exista una diferencia considerable entre el tiempo máximo y el mínimo en la elaboración del modelo de contenido, la dispersión no fue significativa en los tiempos empleados con respecto al tiempo promedio, es decir, 11 minutos y 54 segundos ( $SD=3$  minutos y 14 segundos). La gran diferencia entre el tiempo máximo y el mínimo en la elaboración del modelo de contenido se debió a que algunos usuarios manifestaron mayor habilidad en el manejo de la herramienta InterArch, lo que les permitió terminar la tarea en un periodo de tiempo muy corto (tiempo mínimo: 8 minutos y 40 segundos), mientras

que los dos usuarios que exigieron mayor tiempo en la consecución de la tarea (18 minutos y 45 segundos, y 17 minutos y 13 segundos) coincidieron casualmente con los dos usuarios que menos tiempo dedicaron en la tarea de segmentación de contenido (3 minutos y 56 segundos, y 2 minutos y 55 segundos). Esto confirma que la segmentación es un paso previo importante para agilizar el proceso de AI. Sin embargo, la gran mayoría de los usuarios terminaron la elaboración del modelo de contenido en un tiempo promedio similar.

La tarea de elaborar el modelo de contenido utilizando la herramienta InterArch tuvo una duración total de 2 horas, 22 minutos y 50 segundos, considerando los 12 usuarios.

En cuanto al estilo de manipulación de los elementos visuales en la elaboración del modelo de contenido, la mayoría de los usuarios (8) generó nuevos elementos de contenido en base a la funcionalidad de seleccionar y arrastrar sobre los elementos de contenido creados (herencia). El resto de los usuarios empleó la funcionalidad de arrastrarlos desde la paleta de elementos de la herramienta InterArch, sin presentar mayores dificultades. Sorprendentemente, la funcionalidad que la mayoría de los usuarios (8) eligió presentaba mayor dificultad que la que se utilizó por la minoría (4). Esto confirma el éxito de la funcionalidad de creación de elementos por herencia, que permite establecer elementos de contenido enlazados y con características similares a elementos de contenido de origen, facilitando y agilizando la forma de crear y enlazar los nuevos elementos.

Además, los usuarios manifestaron la necesidad de incluir la funcionalidad mediante la cual los elementos visuales se pudieran organizar automáticamente, en base a su jerarquía y dependencia entre ellos. Es importante señalar que actualmente la herramienta es capaz de organizar los elementos de enlace de forma automática, lo que facilita, en parte, lo requerido por los usuarios. No obstante, la opción de organizar los elementos visuales del diagrama de forma automática es un aspecto importante a tener en cuenta en futuras mejoras de la herramienta InterArch.

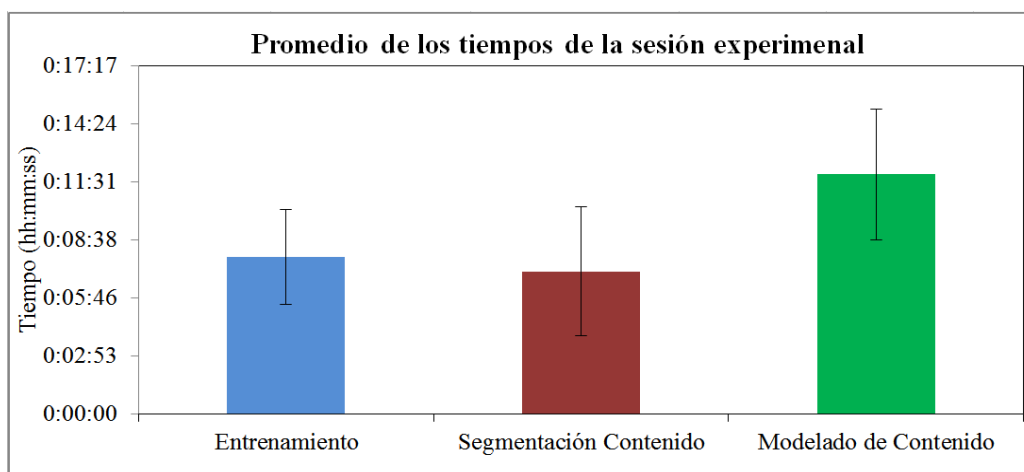
En cuanto al control de errores en la manipulación e interacción con la herramienta, se destaca que la gran mayoría de los usuarios hizo uso de la funcionalidad de deshacer acciones mediante la combinación de las teclas Ctrl+Z, cada vez que querían volver al estado anterior. La combinación de las teclas fue la opción preferida por los usuarios, frente a la opción de edición de la herramienta. Por lo tanto, los resultados demuestran que los usuarios consideraron muy útil esta funcionalidad, ya que están acostumbrados a utilizarla independientemente del entorno de trabajo en el que se encuentren interactuando.

Es importante indicar que la sesión completa de interacción con todos los usuarios (introducción a la herramienta, segmentación del contenido y modelado del contenido utilizando la herramienta InterArch) duró un total de 5 horas, 21 minutos y 33 segundos. El



mayor tiempo que uno de los usuarios requirió fueron 30 minutos y 51 segundos, mientras que el usuario más rápido tardó 19 minutos y 44 segundos.

La etapa de la sesión experimental con menos tiempo requerido fue la **segmentación del contenido**, con una duración total de 1 hora, 25 minutos y 2 segundos, seguida por la **demostración de la herramienta InterArch**, con una duración total de 1 hora, 33 minutos y 41 segundos. La etapa de **crear el modelado de contenido utilizando la herramienta InterArch** fue la más costosa en términos de tiempo, con una duración total de 2 horas, 22 minutos y 50 segundos. No obstante, tal resultado se ha esperado, ya que la etapa de creación del modelado de contenido exigía manipular e interactuar con la herramienta InterArch, en base al conocimiento adquirido en las etapas anteriores, exigiendo, por lo tanto, bastante más tiempo. Así pues, la Figura 81 muestra el promedio de los tiempos (hh:mm:ss) empleados en la sesión experimental, así como una barra correspondiente (+/-  $\sigma$ ) a la dispersión de cada tarea (desviación estándar). La sesión completa de interacción con todos los usuarios tuvo una duración promedio de 26 minutos y 48 segundos (SD=3 minutos y 52 segundos).



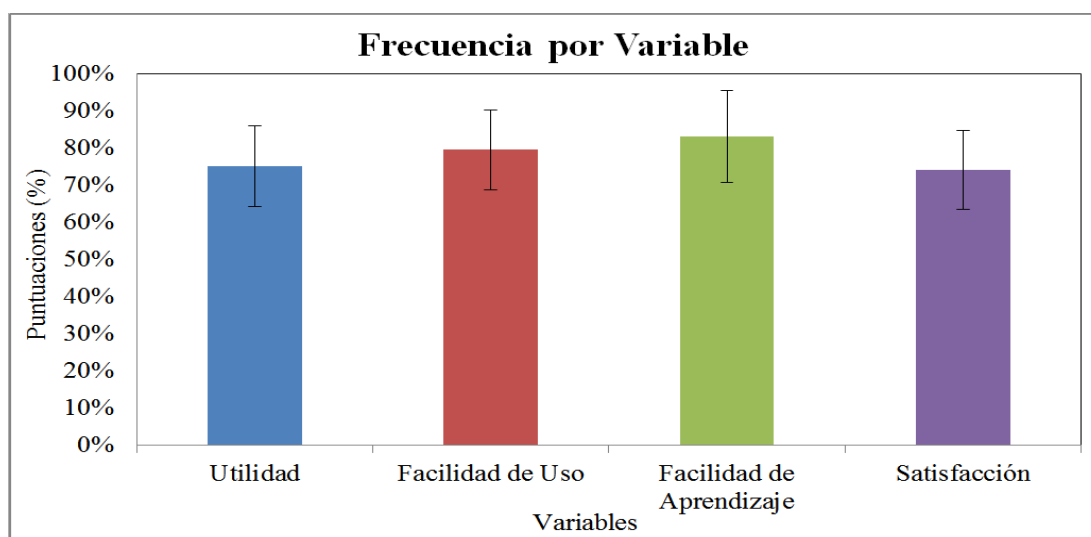
**Figura 81:** Tiempo (hh:mm:ss) promedio empleado por los usuario en la sesión experimental (eficiencia).

El uso del protocolo Thinking Aloud ha permitido obtener información valiosa respecto a cómo los usuarios utilizan la herramienta InterArch, identificando conceptos erróneos a través de los usuarios. Asimismo, un análisis más exhaustivo de las interacciones con la herramienta InterArch fue facilitado gracias a la técnica de evaluación Test Retrospectivo, mediante la revisión de las grabaciones de audio y video de los usuarios. Los resultados obtenidos a través de estas técnicas de evaluación se utilizarán para mejorar la herramienta, para que pueda responder de forma más eficiente, y de acuerdo a las necesidades de los usuarios.

### 5.2.3.2. Resultados de la Usabilidad

Una vez finalizada la sesión experimental, se les solicitó a los usuarios que completaran un cuestionario para medir la usabilidad de la herramienta InterArch. Este cuestionario se diseñó con 30 preguntas cerradas (ver el Anexo A para mayor información) para medir las variables de *Utilidad*, *Facilidad de Uso*, *Facilidad de Aprendizaje* y *Satisfacción*. Además, se incluyeron cuatro preguntas abiertas, con el objetivo de obtener información respecto al perfil de los usuarios, así como los aspectos positivos y negativos de la herramienta InterArch.

La Figura 82 muestra los resultados obtenidos en el análisis del cuestionario, indicando la puntuación media (normalizada en valores entre 0 y 100%), obtenida para cada variable, y una barra de error correspondiente a la desviación estándar ( $\pm \sigma$ ); la media general (línea de puntos horizontal) también se representa.



**Figura 82:** Usabilidad de InterArch. Puntuación media para cada variable que se mide en valores normalizados entre 0 y 100%, incluyendo las barras de error ( $\pm \sigma$ ) y una línea de puntos que representa la media.

Tal y como se muestra en la Figura 82, la dimensión *Facilidad de Aprendizaje* es la que obtiene el promedio de la puntuación más alto de toda la muestra, con una media de 83,05% (SD=12,3%), que corresponde también a la dispersión más alta entre las variables medidas. La variable *Facilidad de Uso* obtiene el segundo promedio más alto de las puntuaciones medidas, con un promedio de 79,5% (SD=10,83%). Le sigue la variable *Utilidad*, que obtiene un promedio de la puntuación de 75% (SD=10,89%). Finalmente, la variable *Satisfacción* fue la que obtuvo el promedio de puntuación más bajo, de 74,16% (SD=10,59%), y que corresponde también a la dispersión más baja entre las variables medidas.

En términos generales, se puede señalar que el promedio de las puntuaciones de toda la muestra es de 77,93% (SD=11,4%). Destaca el hecho que todas las dimensiones obtienen

puntuaciones superiores a 74%. Por lo tanto, esta medida se puede considerar como un buen indicador de la usabilidad general de la herramienta InterArch, en base a la percepción de los usuarios.

#### **5.2.3.3. Fiabilidad de la Evaluación**

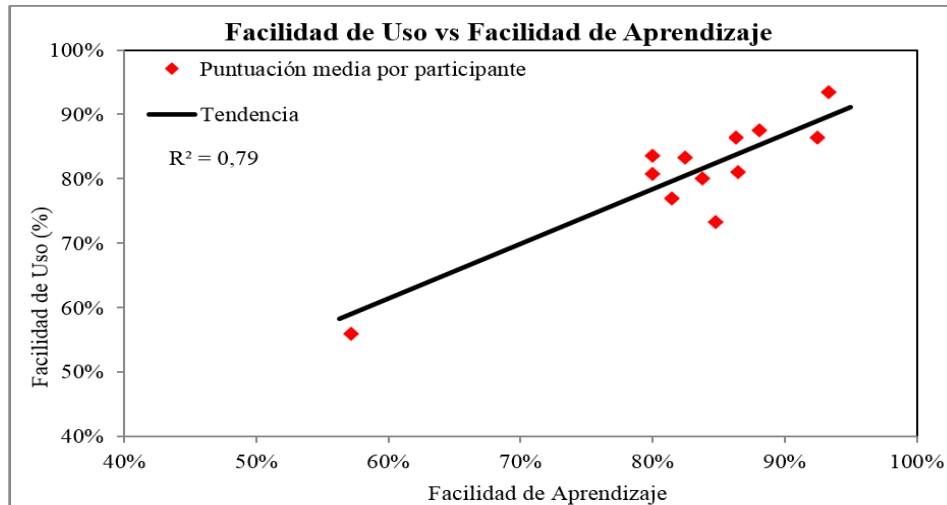
La fiabilidad de la evaluación permite validar el nivel de consistencia de una determinada medida, que debería mostrar resultados similares, libres de error, en repetidas mediciones. Para medir la fiabilidad de la evaluación realizada a los 12 usuarios, se ha utilizado el indicador Alfa de Cronbach. Este indicador determina la consistencia interna de una escala, y toma valores entre 0 y 1; en ocasiones, toma valores negativos, en los casos donde hay ítems que miden lo opuesto al resto. Cuanto más se acerque el valor calculado a 1, más alta será la fiabilidad de la evaluación, considerándose como un nivel de fiabilidad aceptable entre los valores 0.7 y 0.8, un nivel bueno entre los valores 0.8 y 0.9, y excelente en valores mayores a 0.9 (Cronbach, 1951).

El indicador de Alfa de Cronbach se ha calculado para las 30 preguntas cerradas del cuestionario. El resultado muestra un valor de fiabilidad del 95.18% ( $\alpha=0.95$ ), indicando que el cuestionario presenta un nivel de fiabilidad excelente, pues supera en un 25.18% el umbral de aceptación.

#### **5.2.3.4. Análisis Multivariable**

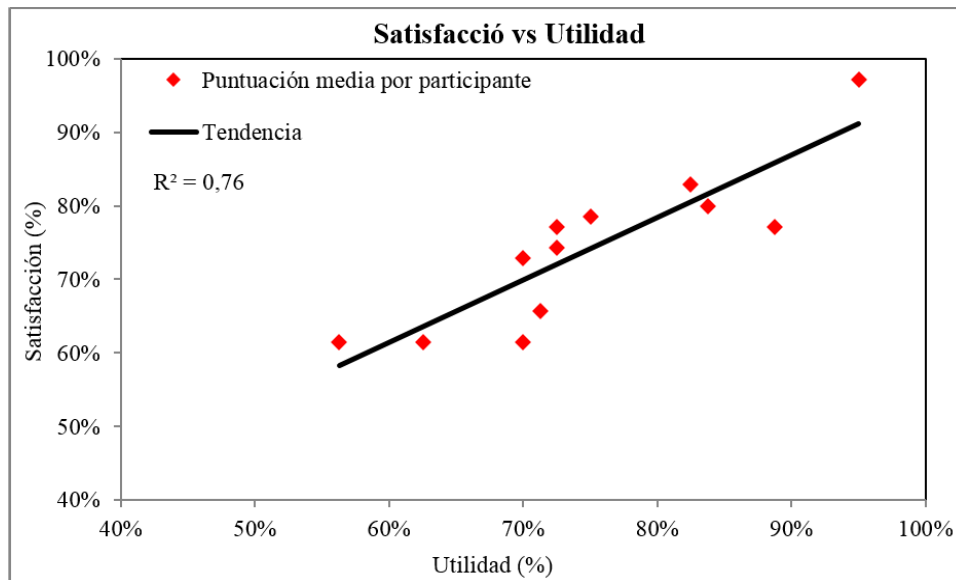
Debido a que los resultados de las medias de las cuatro dimensiones están próximos entre sí, se estudió la posible correlación de tales valores. Así pues, se realizó un cálculo para medir el coeficiente de correlación entre las dimensiones evaluadas. Este análisis tiene como objetivo medir el grado de intensidad lineal de vinculación de las dimensiones. Para llevar a cabo este cálculo, primero se elaboró una matriz de correlación para estudiar la correlación de todas las dimensiones y así poder seleccionar las que estaban más correlacionadas (con un umbral del coeficiente de correlación superior al 70%). A continuación, se efectuaron gráficos de dispersión de las puntuaciones promedio de cada una de las dimensiones seleccionadas, confrontando unas puntuaciones con otras y calculando el coeficiente de correlación lineal ( $R^2$ ). Este coeficiente se sitúa entre 0 y 1, habiendo una mayor relación lineal entre las dimensiones al acercarse al 1.

La correlación más alta encontrada fue entre las dimensiones *Facilidad de Uso* y *Facilidad de Aprendizaje*, con un coeficiente de correlación lineal en sentido positivo de  $R^2=0.79$ . Como se puede observar en la Figura 83, sólo un usuario ha entregado una puntuación neutra respecto a la conformidad de estas dos dimensiones. Esto se corresponde con el mismo usuario que en promedio entregó la puntuación más baja en todas las dimensiones, con un promedio de la puntuación de 53,9%. Sin embargo, tal y como se muestra en el gráfico, la mayor parte de los usuarios manifestó niveles altos de conformidad en cuanto a las cualidades de facilidad de uso de la herramienta, que se ve incrementada en la medida en que la herramienta ofrece mayores facilidades en el aprendizaje de su utilización.



**Figura 83:** Correlación entre las dimensiones *Facilidad de uso* y *Facilidad de Aprendizaje*.

La Figura 84 muestra la segunda correlación más alta que se encontró entre las dimensiones *Satisfacción* y *Utilidad*, obteniendo un coeficiente de correlación lineal de  $R^2=0,76$ , lo que demuestra que existe una correlación en sentido positivo entre estas dimensiones. Esta correlación permite indicar que los usuarios manifiestan mayor satisfacción en la utilización de la herramienta, en la medida que ésta cumple con las expectativas de utilidad.



**Figura 84:** Correlación entre las dimensiones *Satisfacción* y *Utilidad*.

#### 5.2.4. Discusión de los Resultados

En esta sección se realiza una discusión de las dos partes de la experimentación llevada a cabo, es decir, la sesión experimental con los 12 usuarios y con el cuestionario de usabilidad.

Respecto al grado de eficacia de las tareas del experimento, los resultados muestran que en todos los casos los usuarios finalizaron las tareas correctamente. Por lo tanto, es posible afirmar que se ha alcanzado un 100% de eficacia.

En cuanto a la eficiencia, los tiempos medios empleados por los usuarios fueron bastante variados. Estos tiempos permiten indicar que los usuarios que menos tiempo emplearon en la tarea de segmentación de contenido fueron los que más tiempo dedicaron en la tarea de crear el modelo de contenido utilizando la herramienta. Esto sólo muestra un determinado estilo de trabajo para llevar a cabo las tareas, pues no fueron los que emplearon ni más ni menos tiempo en toda la sesión de interacción.

En cuanto a la satisfacción, los resultados en las cuatro dimensiones de usabilidad arrojan valoraciones más que aceptables, puesto que en todas ellas se obtienen puntuaciones sobre 74% ( $M=77,9\%$ ,  $11,4\%$ ). Estos resultados de las dimensiones, junto con el grado de eficacia logrado en las tareas del experimento, permiten confirmar que InterArch provee funcionalidades para la creación de modelos de contenido que son fáciles de utilizar y de aprender por parte de los usuarios finales.

Con respecto a las tomas de audio y de video durante las sesiones experimentales, éstas han permitido obtener información valiosa respecto a cómo los usuarios utilizan la herramienta InterArch. Por ejemplo, aunque los usuarios optaron mayoritariamente por utilizar la funcionalidad de seleccionar y arrastrar sobre los elementos de contenido creados (herencia), se detectó que esta funcionalidad presenta cierta dificultad en su manipulación, y que los usuarios necesitaban efectuar varios intentos antes de conseguir lo que querían de forma exitosa. Asimismo, se observó que los usuarios realizaron varios pasos en la búsqueda de las distintas opciones de la herramienta. Esto permitió detectar la necesidad de contar con más opciones de acceso directo y de definir una organización que permita conseguir una consecución más ágil, en función del número de pasos necesarios para alcanzar las opciones que requieren los usuarios.

En cuanto a las preguntas abiertas del cuestionario, éstas reportaron información valiosa respecto a los aspectos positivos y de mejora para la herramienta. Si bien estas preguntas fueron presentadas a los usuarios de forma opcional, destaca que todos los usuarios se interesaron en realizar observaciones y entregar comentarios respecto a su experiencia. En cuanto a los comentarios de los aspectos positivos observados, cabe comentar que los usuarios resaltaron las siguientes propiedades: sencillez, facilidad de uso, rapidez, utilidad, intuición, amigabilidad, sentido práctico y parecido con otros entornos, entre otras. Esto permite corroborar las evaluaciones positivas obtenidas en las preguntas cerradas, que, además, están en concordancia con los aspectos que se plantearon al desarrollar la herramienta. Respecto a los aspectos susceptibles de mejora que requeriría la herramienta, los usuarios aportaron información valiosa a través de los siguientes comentarios: *“faltan tipos de elementos para utilizar”*, *“es muy sensible al mouse”*, *“las cajas se deberían representar en distintos tamaños, según el nivel de jerarquía en el modelo de contenido”*,

“el texto debería adaptarse al tamaño de la caja”, y “falta una organización automática del diagrama completo”. Todos estos comentarios reportaron una valiosa información de cara a la mejora continua y serán considerados en futuras mejoras de la herramienta InterArch.

Asimismo, los resultados obtenidos proporcionan evidencia suficiente para responder las preguntas de la investigación indicadas anteriormente:

- **PI18:** ¿Cuál es la eficacia alcanzada por los usuarios en la realización de un modelo de contenido utilizando InterArch?

La eficiencia alcanzada es positiva en todos los casos, los resultados muestran que los usuarios finalizaron las tareas correctamente.

- **PI19:** ¿Cuál es la eficiencia alcanzada por los usuarios en la realización de un modelo de contenido utilizando InterArch?

Tras haber revisado los resultados de los tiempos medios utilizados en las tareas de segmentación del contenido, y elaboración del modelado de contenido utilizando la herramienta InterArch, se puede indicar que éstos son razonables, debido a que rondan los 10 minutos, lo que es esperable dadas las acciones solicitadas a los usuarios.

- **PI20:** ¿Cuál es la percepción de usabilidad de los usuarios con respecto a la herramienta InterArch?

La percepción de usabilidad de los 12 Arquitectos de la Información con respecto a la herramienta InterArch es positiva en todos los casos. En estos resultados, se ha obtenido una puntuación media de 77,93% (SD=11,4%) y destaca la alta valoración obtenida para las dimensiones *Facilidad de Aprendizaje* y *Facilidad de Uso*, con puntuaciones medias de 83,05% (SD=12,3%) y 79,5% (SD=10,83), respectivamente.

Finalmente, todos los resultados anteriores permiten corroborar la hipótesis de partida **H2.3** al concluir que InterArch es fácil de usar y de aprender por parte del Arquitecto de la Información, y que, además, permite generar información de análisis y diseño para Analistas e Ingenieros del Software de forma automática, a partir de modelos de contenido que el Arquitecto de la Información crea, facilitando así el desarrollo incremental y ágil en Scrum-UIA.

### 5.3. Evaluación de la Usabilidad de Scrum-UIA-MAT

Una vez presentadas las evaluaciones del método de priorización QMPSR y de la herramienta InterArch, en esta sección se describe la experimentación llevada a cabo para medir el grado de usabilidad alcanzado en la herramienta Scrum-UIA-MAT. Para la validación de la usabilidad de dicha herramienta, se ha realizado un diseño cuasi experimental con 16 Ingenieros del Software. En primer lugar, en la sección 5.3.1, se presentan las variables, preguntas de investigación, participantes y tareas de la experimentación. Posteriormente, en la sección 5.3.2, se describen los detalles de la ejecución del experimento. Seguidamente, en la sección 5.3.3, se describen y analizan los diferentes resultados obtenidos de la sesión experimental con los 16 usuarios y del cuestionario de usabilidad. Finalmente, en la sección 5.3.4, se realiza una discusión respecto a los resultados generales obtenidos de la experimentación.

Asimismo, con la realización de esta evaluación, también se busca corroborar la hipótesis de partida **H2.4**: Es posible crear una herramienta Web, que proporcione un sistema de gestión de proyectos basado en la metodología anteriormente ideada (hipótesis de partida **H2.1**), mediante un proceso ingenieril centrado en el usuario. De este modo, la herramienta Web desarrollada será fácil de usar y de aprender por parte de sus usuarios potenciales.

#### 5.3.1. Marco Experimental

El objetivo principal de la experimentación fue estudiar y analizar el grado de usabilidad de Scrum-UIA-MAT. Debido a que no se realizó una selección aleatoria de los grupos de estudio, se ha efectuado un diseño cuasi experimental, es decir, se ha escogido el grupo de estudio dentro del entorno próximo al investigador.

Al igual que la evaluación de la herramienta InterArch, este experimento se ha diseñado con el objetivo de cumplir con tres condiciones de usabilidad (eficacia, eficiencia y satisfacción), descritas en el estándar ISO-9241-11 (DIN, 1998). De esta manera, se espera que los resultados de la evaluación de la usabilidad de la herramienta Scrum-UIA-MAT muestren niveles aceptables, y que los usuarios perciban la herramienta con una navegación consistente e intuitiva. Para lograr lo anterior, se ha diseñado y seleccionado un conjunto de 12 tareas, con el fin de completar un ciclo para gestionar un proyecto. Estas tareas se centran en encontrar y completar la información que es esencial para gestionar un proyecto, restando elementos de los que normalmente se puede prescindir.

##### 5.3.1.1. Variables y Preguntas de Investigación

Las variables y preguntas de investigación se han definido de acuerdo al objetivo de la experimentación. De esta manera, las variables independientes fueron el tiempo de asistencia requerido por los usuarios en la realización de las tareas, el número de tareas efectuadas por los usuarios, el tiempo total requerido por los usuarios para acabar las tareas, los rasgos de los usuarios (edad, género y conocimiento en informática), y las respuestas de los usuarios a las preguntas de un cuestionario de usabilidad.

Respecto a las variables dependientes, éstas se definen en base a las tres condiciones de usabilidad, indicadas en el estándar ISO-9241-11 (eficacia, eficiencia y satisfacción). De esta forma, la eficacia se refiere al grado de éxito que consiguen los usuarios en la ejecución de las tareas, y se mide en términos de porcentaje de consecución de las tareas por parte de los usuarios. La eficiencia consiste en el tiempo requerido por los usuarios para completar cada tarea, y se mide en los segundos que los usuarios necesitan para cumplir cada una de ellas. Por último, la satisfacción es la percepción subjetiva sobre la usabilidad conseguida por la herramienta Scrum-UIA-MAT, y se mide a través del valor medio de las respuestas de los usuarios a las preguntas del cuestionario de usabilidad.

Finalmente, se han definido las preguntas de investigación, con el fin de responder a las tres variables dependientes (eficacia, eficiencia y satisfacción). De esta manera, las preguntas de investigación (presentadas en la sección 1.2) corresponden a:

**PI<sub>21</sub>:** ¿Cuál es la eficacia alcanzada por los usuarios en la realización de tareas de gestión de un proyecto utilizando Scrum-UIA-MAT?

**PI<sub>22</sub>:** ¿Cuál es la eficiencia alcanzada por los usuarios en la realización de tareas de gestión de un proyecto utilizando Scrum-UIA-MAT?

**PI<sub>23</sub>:** ¿Cuál es la percepción de usabilidad de los usuarios con respecto a Scrum-UIA-MAT?

### 5.3.1.2. Participantes

En la evaluación de la herramienta Scrum-UIA-MAT participaron 16 personas que trabajan o estudian en temas afines a informática, específicamente en tareas relacionadas con la gestión de proyectos software en entornos ágiles, así como en el uso de herramientas CASE. En la evaluación participaron 11 hombres y 5 mujeres, con edades comprendidas entre 23 y 57 años ( $M=33$ ,  $SD=10,06$ ). Cabe destacar la alta dispersión en las edades de los participantes de la evaluación, lo que permite afirmar que la muestra de la población no estuvo concentrada en ningún estrato específico de edades. Por lo tanto, esta diversidad en las edades de los participantes de la evaluación se considera como un elemento positivo para llevar a cabo la evaluación de la herramienta Scrum-UIA-MAT.

En cuanto al nivel de conocimiento de los usuarios sobre educación y gestión de requisitos, así como de metodologías de desarrollo de software tanto tradicionales como ágiles, esta medición se llevó a cabo a través de una escala Likert de 1 a 5, donde 1 equivalía a un nivel *Nulo*, y 5 a un nivel *Experto*. El nivel medio resultante entre los participantes fue 4,06 ( $SD=0,68$ ).

### 5.3.1.3. Tareas

Para llevar a cabo el experimento, se pidió a cada usuario realizar 12 tareas para configurar un proyecto a través del uso de la herramienta Scrum-UIA-MAT. Específicamente, las 12 tareas que realizaron los usuarios corresponden a:



- **Crear un proyecto (T1):** El usuario debe ingresar un nuevo proyecto, con el fin de gestionar y configurar todos sus datos.
- **Agregar usuarios finales del proyecto (T2):** Se solicita al usuario ingresar los usuarios finales del proyecto.
- **Agregar aspectos relevantes del proyecto (T3):** Se solicita al usuario ingresar los aspectos relevantes del proyecto.
- **Agregar elementos de los aspectos relevantes del proyecto (T4):** Se solicita al usuario ingresar diferentes elementos, relacionados con los aspectos relevantes del proyecto.
- **Configurar la prioridad de los aspectos relevantes del proyecto (T5):** Se solicita al usuario asignar una prioridad a cada aspecto relevante del proyecto.
- **Agregar un requisito en la Lista del Producto del proyecto (T6):** Se solicita al usuario ingresar un requisito en el proyecto.
- **Ordenar la Lista del Producto (T7):** Se solicita al usuario ordenar la Lista del Producto, de acuerdo a la prioridad del aspecto relevante relacionado con la “Usabilidad”.
- **Agregar un Sprint (T8):** Se solicita al usuario ingresar un Sprint inicial en el proyecto y especificar una duración aproximada.
- **Asignar un requisito a un Sprint (T9):** Se solicita al usuario asignar un requisito al Sprint inicial del proyecto.
- **Agregar una tarea de desarrollo (T10):** Se solicita al usuario ingresar una tarea de desarrollo en algún requisito del Sprint inicial del proyecto, especificando datos relacionados con la estimación y la descripción de la tarea de desarrollo.
- **Consultar las tareas de desarrollo (T11):** Se solicita al usuario consultar las tareas de desarrollo del Sprint inicial del proyecto.
- **Ingresar una estimación de las tareas de desarrollo (T12):** Se solicita al usuario ingresar una estimación del esfuerzo diario en las tareas de desarrollo del Sprint inicial del proyecto.

Las instrucciones proporcionadas a cada participante de la experimentación se pueden ver en el Anexo B.

### 5.3.2. Ejecución de la Experimentación

La experimentación se ha llevado a cabo de forma presencial o remota, dependiendo de la disponibilidad de los usuarios. En el modo presencial, se acordaba un lugar físico para efectuar la reunión, mientras que en el modo remoto se efectuaba una videoconferencia utilizando Skype y el escritorio remoto de Chrome, con el fin de que los usuarios pudieran compartir su pantalla y de esta forma comprobar los pasos realizados. En ambos casos (presencial y remoto) se realizaron los siguientes pasos:

1. **Primera Etapa:** Inicialmente, a cada participante de la evaluación se le efectuó una introducción al objetivo general de la evaluación y a los conceptos de la metodología Scrum-UIA, utilizados para desarrollar la herramienta Scrum-UIA-MAT. Esta introducción tuvo una duración promedio de 8 minutos y 33 segundos (SD=1 minuto y 26 segundos).
2. **Segunda Etapa:** Una vez finalizada la introducción de la evaluación de la herramienta Scrum-UIA-MAT, y como parte de la tarea a realizar en el experimento, se requirió a cada uno de los participantes realizar 12 tareas relacionadas con la configuración de un proyecto, utilizando la herramienta Scrum-UIA-MAT (estas tareas se han presentado en términos generales en la sección 5.3.1.3, y su descripción completa se encuentra en el Anexo B). La ejecución de la herramienta se llevó a cabo en el navegador de internet que los usuarios seleccionaron, con el fin de que la experimentación se simulara en el entorno de trabajo habitual de los usuarios. Asimismo, a los usuarios se les solicitó que indicaran, en voz alta, los pensamientos que tenían durante el uso de la herramienta, con el fin de aplicar el protocolo Thinking Aloud. Esto proporcionó información sobre las interacciones efectuadas con la herramienta Scrum-UIA-MAT durante la realización de la tarea.
3. **Tercera Etapa:** En el último paso de la experimentación, y una vez finalizada la sesión con los usuarios, se les solicitó a cada uno de los participantes rellenar un cuestionario para evaluar la usabilidad de la herramienta Scrum-UIA-MAT. Este cuestionario es el mismo que se utilizó para evaluar la usabilidad de la herramienta InterArch (ver Anexo A).

### 5.3.3. Resultados Obtenidos

En esta sección se detallan y analizan los resultados obtenidos de la sesión experimental con los 16 usuarios (Primera y Segunda Etapas) y del cuestionario de usabilidad (Tercera Etapa).

#### 5.3.3.1. Resultados de la Sesión Experimental (Primera y Segunda Etapas)

En esta sección se describen los diferentes resultados de la sesión experimental, que se muestran de acuerdo al orden de las actividades establecidas para el proceso de evaluación de la herramienta Scrum-UIA-MAT:

- c) **Primera Etapa:** Esta etapa tuvo una duración promedio de 8 minutos y 33 segundos (SD=1 minutos y 26 segundos), con una duración total (con los 16 usuarios) de 2 horas, 16 minutos y 48 segundos. El tiempo máximo requerido por uno de los usuarios fueron 11 minutos y 18 segundos, mientras que el tiempo mínimo fueron de 5 minutos y 51 segundos. Las diferencias entre los tiempos máximos y mínimos, indicados anteriormente, se deben principalmente a que hubo usuarios que conocían la nomenclatura que se utilizaba normalmente en metodologías ágiles, y que apenas realizaron preguntas. Mientras tanto, en otros

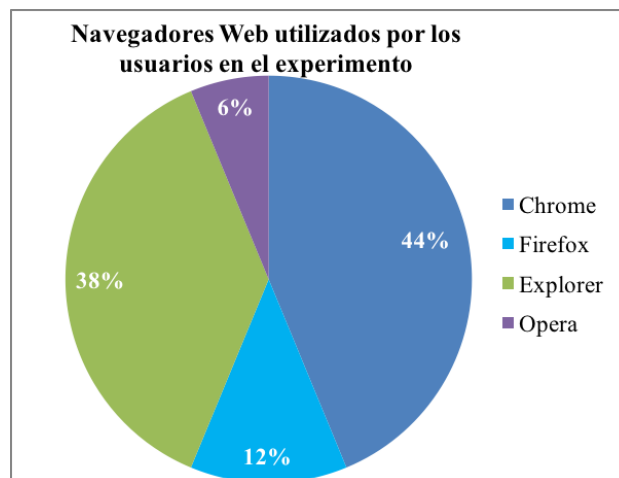
casos había usuarios que desconocían algunos conceptos pilares de las metodologías ágiles (por ejemplo: Lista del Producto, Sprint, Tarea de Desarrollo, entre otros), lo que exigió extender el tiempo de esta etapa, con el fin de aclarar estos conceptos.

Como se mencionó anteriormente, esta etapa tuvo como objetivo principal presentar a los usuarios algunos conceptos de la metodología Scrum-UIA y de los métodos ágiles. Algunos de estos conceptos corresponden a:

- Los requisitos se agrupan a través de la Lista del Producto.
- La priorización de los requisitos se conduce por los aspectos y elementos relevantes del proyecto.
- El Dueño del Producto y el Arquitecto de la Información pueden priorizar los requisitos de manera independiente.
- Los requisitos se describen en formato de historias de usuarios, con el fin de identificar a los usuarios finales, sus objetivos y el valor de negocio esperado.
- Las tareas de desarrollo se agrupan en los Sprints.
- La estimación del esfuerzo diario de las tareas de desarrollo se realiza por Sprint.

**d) Segunda Etapa:** Esta etapa consistió en proporcionar a los usuarios un enunciado que contenía la descripción de 12 tareas para configurar algunos aspectos de un proyecto, utilizando la herramienta Scrum-UIA-MAT (ver Anexo B para su descripción completa).

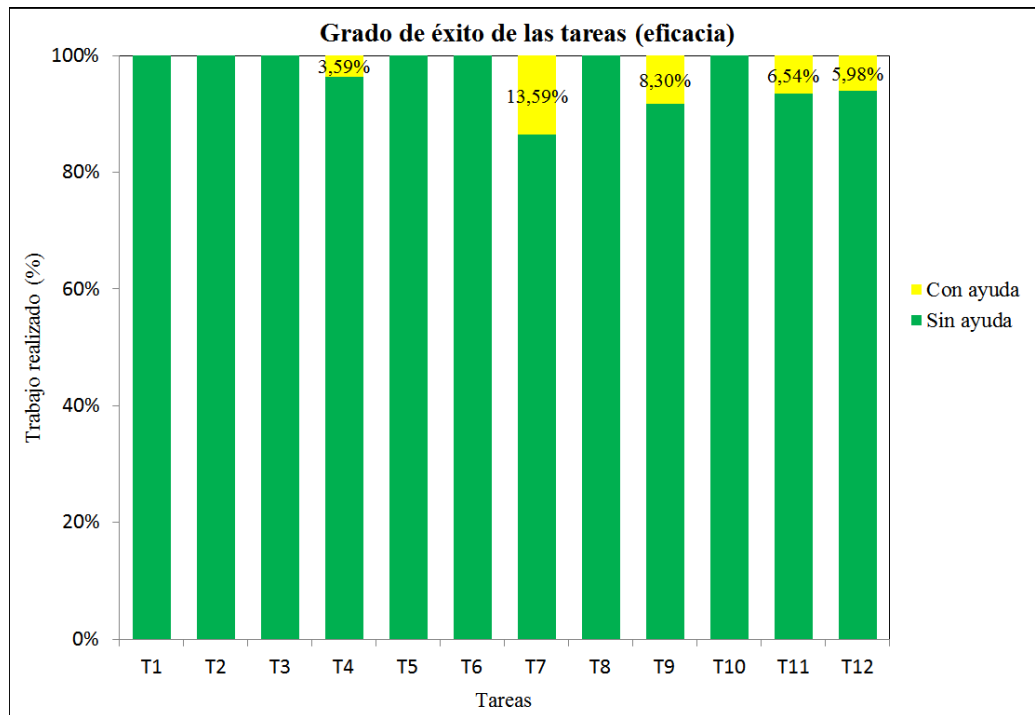
En la Figura 85, se presentan los diferentes navegadores Web, utilizados por los usuarios en el experimento. Como se puede observar, los navegadores más usados fueron Chrome y Explorer con un 44% y 38%, respectivamente. Estos son seguidos por los navegadores Firefox y Opera con un 12% y 6%, respectivamente.



**Figura 85:** Navegadores Web utilizados por los usuarios en el experimento.

El primer punto medido ha sido la eficacia obtenida en la realización de las distintas tareas, calculada a través del grado de éxito en la finalización de las tareas del experimento por parte de los usuarios. Los resultados muestran que en todos los casos los usuarios finalizaron las tareas correctamente. Por lo tanto, se puede indicar que se ha alcanzado un 100% de eficacia.

Adicionalmente, se ha distinguido entre los casos en los que las tareas se ejecutaron sin ninguna consulta o duda por parte de los usuarios (denominado “Sin ayuda”), y los casos en los que los usuarios tenían consultas y era necesario responder a sus preguntas durante la ejecución (denominado “Con ayuda”). Los resultados de la eficacia de cada tarea del experimento se muestran en el gráfico de la Figura 86.



**Figura 86:** Grado de éxito en la finalización de las tareas del experimento por parte de los usuarios (eficacia).

Como se puede observar en la Figura 86, solamente en las tareas T4, T7, T9, T11 y T12 fue necesario responder a preguntas durante su ejecución. Las tareas T7, T9 y T11 fueron las que más ayuda demandaron para completarse, con una asistencia media de 13,59% (SD=1,88), 8,3% (SD=1,6%) y 6,54 (SD=1,12%), respectivamente. Este porcentaje de asistencia (ver color amarillo en la Figura 86) fue medido como el tiempo utilizado para completar la tarea con ayuda, respecto al tiempo total utilizado en la tarea.

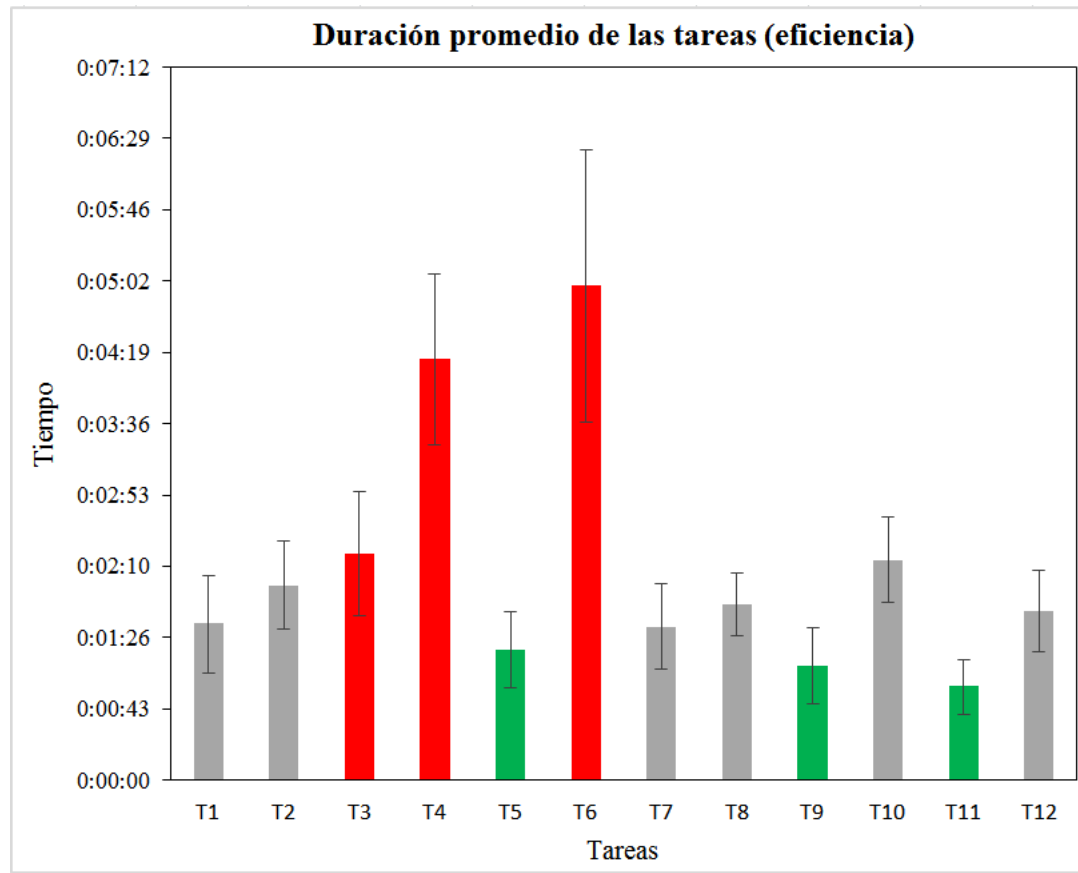
En la tarea T7, los usuarios debían ordenar los requisitos, de acuerdo a un aspecto relevante del proyecto. En este caso, tres usuarios tuvieron dudas porque el módulo ofrecía más alternativas de las que ellos esperaban. Respecto a la tarea T9, ésta

consistía en asignar un requisito a un Sprint, donde un usuario tuvo problemas para encontrar el botón que permite hacerlo, mientras que otro usuario no seleccionó correctamente el requisito para llevar a cabo la asignación. Finalmente, en la tarea T11, los usuarios debían consultar las tareas de desarrollo de un Sprint, donde dos usuarios tuvieron inconvenientes para identificar la opción que permite listar dichas tareas.

El siguiente punto medido corresponde a la eficiencia conseguida en la elaboración de las distintas tareas. Este punto fue medido como el tiempo requerido por los usuarios para llevar a cabo las tareas. En la Tabla 31, se presentan las estadísticas de tiempo (hh:mm:ss) utilizado por los usuarios en la ejecución de las tareas en el experimento. Asimismo, se ha elaborado la Figura 87 para presentar, de manera gráfica, el detalle de los tiempos (hh:mm:ss) promedios, utilizados en cada una de las 12 tareas, con barras de error con un nivel de confianza del 95%.

Tareas	Media	Error Típ.	Intervalo de confianza para la media al 95%		Desv. Est.	Mín.	Máx.
			Límite inferior	Límite superior			
T1	0:01:35	0:00:13	0:01:05	0:02:04	0:00:55	0:00:28	0:03:26
T2	0:01:58	0:00:12	0:01:31	0:02:25	0:00:50	0:00:46	0:03:45
T3	0:02:17	0:00:17	0:01:40	0:02:55	0:01:10	0:00:22	0:04:12
T4	0:04:15	0:00:24	0:03:23	0:05:07	0:01:37	0:01:55	0:07:43
T5	0:01:19	0:00:10	0:00:56	0:01:42	0:00:43	0:00:21	0:03:08
T6	0:05:00	0:00:38	0:03:37	0:06:22	0:02:34	0:02:01	0:09:57
T7	0:01:33	0:00:12	0:01:07	0:01:59	0:00:48	0:00:28	0:03:12
T8	0:01:46	0:00:08	0:01:27	0:02:05	0:00:35	0:00:39	0:02:46
T9	0:01:09	0:00:10	0:00:46	0:01:32	0:00:43	0:00:21	0:02:51
T10	0:02:13	0:00:12	0:01:47	0:02:39	0:00:48	0:01:15	0:04:06
T11	0:00:57	0:00:07	0:00:40	0:01:13	0:00:30	0:00:12	0:02:10
T12	0:01:42	0:00:11	0:01:17	0:02:07	0:00:46	0:00:29	0:03:25

**Tabla 31:** Estadísticas del tiempo (hh:mm:ss) de ejecución de las tareas en el experimento.



**Figura 87:** Tiempo (hh:mm:ss) promedio de duración de las tareas de la experimentación (eficiencia).

Como se puede observar, los usuarios tardaron más tiempo en las tareas T6, T4 y T3, que tuvieron una duración promedio de 5 minutos (SD=2 minutos y 34 segundos), 4 minutos y 15 segundos (SD=1 minuto y 37 segundos), y 2 minutos y 17 segundos (SD=1 minuto y 10 segundos), respectivamente. Estas tareas tienen en común el hecho de que requerían ingresar una mayor cantidad de datos en comparación con otras tareas, lo que produjo como resultado que los usuarios destinaran más tiempo para digitar estos datos.

Por el contrario, los usuarios tardaron menos tiempo en las tareas T11, T9 y T5, que tuvieron una duración promedio de 57 segundos (SD=30 segundos), 1 minuto y 9 segundos (SD=43 segundos) y 1 minutos y 19 segundos (SD=43 segundos), respectivamente. En estas tareas, los usuarios solamente debían realizar acciones referentes a encontrar ciertas funcionalidades de la herramienta (consultar las tareas de desarrollo, asignar un requisito en el Sprint y configurar la prioridad de los aspectos correspondientes a las tareas T11, T9 y T5, respectivamente), y no requerían digitar ningún tipo de información, generando, como resultado, que los usuarios destinaran menos tiempo.

Respecto a los valores del intervalo de confianza (ver Tabla 31), todas las tareas tienen valores inferiores a un minuto, con excepción de la tarea T6 (1 minuto y 22 segundos). Estos resultados permiten caracterizar los tiempos que van a requerir los usuarios en la realización de las tareas. Por ejemplo, en la tarea T1, se puede estar 95% seguro de que un usuario necesitará entre 1 minuto y 5 segundos y 2 minutos y 4 segundos para agregar un nuevo proyecto en la herramienta. Por lo tanto, estos valores del intervalo de confianza se consideran como aceptables.

La realización completa de las tareas utilizando la herramienta Scrum-UIA-MAT tuvo una duración promedio de 25 minutos y 49 segundos ( $SD=8$  minutos y 4 segundos). El tiempo máximo requerido por uno de los usuarios fueron 43 minutos y 44 segundos, mientras que el tiempo mínimo fueron 10 minutos y 18 segundos.

En general, las diferencias de los tiempos utilizados por los usuarios se deben principalmente a las diferentes habilidades que presentaban éstos para manipular, digitar y realizar las diferentes tareas propuestas.

Un aspecto a destacar es que los usuarios manifestaron la necesidad de que la herramienta incluyera alguna funcionalidad que permitiera obtener retroalimentación al finalizar la incorporación de nuevos requisitos en la Lista del Producto (tarea T6). Cabe mencionar que actualmente la herramienta posee una funcionalidad que genera retroalimentación en los casos de errores de digitación o de registro, al momento de incorporar nuevos requisitos a la Lista del Producto. No obstante, esta funcionalidad manifestada por los usuarios es un aspecto importante a considerar en futuras mejoras de la herramienta Scrum-UIA-MAT.

En cuanto a la estructura del diseño de la herramienta Scrum-UIA-MAT (organización y estructura de las funcionalidades), los usuarios navegaron sin mayores inconvenientes por sus diferentes opciones para configurar y gestionar los diferentes aspectos del proyecto. Solamente en ciertos casos, algunos usuarios buscaban más de una opción para intentar llevar a cabo las tareas propuestas, que en su mayoría correspondían a alguna de las tareas iniciales. No obstante, los usuarios no se mostraron desorientados por la estructura de diseño propuesta por la herramienta.

En términos generales, se puede mencionar que la sesión experimental completa de interacción con los 16 usuarios (primera y segunda etapas) tuvo una duración total de 9 horas, 9 minutos y 59 segundos, con una duración promedio de 34 minutos y 22 segundos ( $SD=8$  minutos y 24 segundos). El tiempo máximo requerido por uno de los usuarios fueron 53 minutos y 41 segundos, mientras que el tiempo mínimo fueron 17 minutos y 34 segundos.

La segunda etapa de la sesión experimental requirió más tiempo que la primera, con una diferencia promedio de 25 minutos y 49 segundos. Lo anterior era de esperar, ya que la

primera etapa no requería un esfuerzo especial, mientras que la realización de las 12 tareas demandaba mayor dedicación de tiempo, dado que exigía manipular e interactuar con la herramienta Scrum-UIA-MAT.

Finalmente, el uso del protocolo Thinking Aloud ha facilitado la obtención de datos cualitativos de los usuarios, respecto al diseño y las funcionalidades de la herramienta. Por ejemplo, dos usuarios esperaban encontrar la funcionalidad para editar las tareas de desarrollo dentro del listado de los requisitos del Sprint. Asimismo, tres usuarios intentaron realizar la asignación del requisito al Sprint, arrastrando el requisito con el cursor al área del Sprint. Estos y los demás datos, obtenidos con el protocolo Thinking Aloud, se utilizarán para incorporar diferentes mejoras en la herramienta, con el fin de que los usuarios puedan cumplir sus objetivos de manera eficiente.

### 5.3.3.2. Resultados y Análisis de la Usabilidad (Tercera Etapa)

Al finalizar la sesión experimental, se les solicitó a los usuarios completar un cuestionario, con el fin de medir la usabilidad de la herramienta Scrum-UIA-MAT. Dicho cuestionario está compuesto por 30 preguntas cerradas y cuatro preguntas abiertas (el Anexo A contiene el cuestionario completo). Por un lado, las 30 preguntas cerradas tienen como objetivo medir la usabilidad de la herramienta a través de las siguientes dimensiones: *Utilidad*, *Facilidad de Uso*, *Facilidad de Aprendizaje* y *Satisfacción*. Por otro lado, las cuatro preguntas abiertas tienen como objetivo obtener detalles sobre el perfil de los usuarios y los aspectos positivos y negativos de la herramienta Scrum-UIA-MAT.

A diferencia de la evaluación de la usabilidad de la herramienta InterArch, en esta evaluación el número de tareas y de usuarios fue mayor. Por lo tanto, se decidió describir y analizar los resultados de las 30 preguntas del cuestionario en detalle, agrupando las preguntas de acuerdo a las diferentes dimensiones medidas (*Utilidad*, *Facilidad de Uso*, *Facilidad de Aprendizaje* y *Satisfacción*). Todas las preguntas cerradas fueron puntuadas por los usuarios con la escala de Likert entre 1 y 7, siendo un 7 el valor máximo, un 1 el valor mínimo y un 4 el valor intermedio.

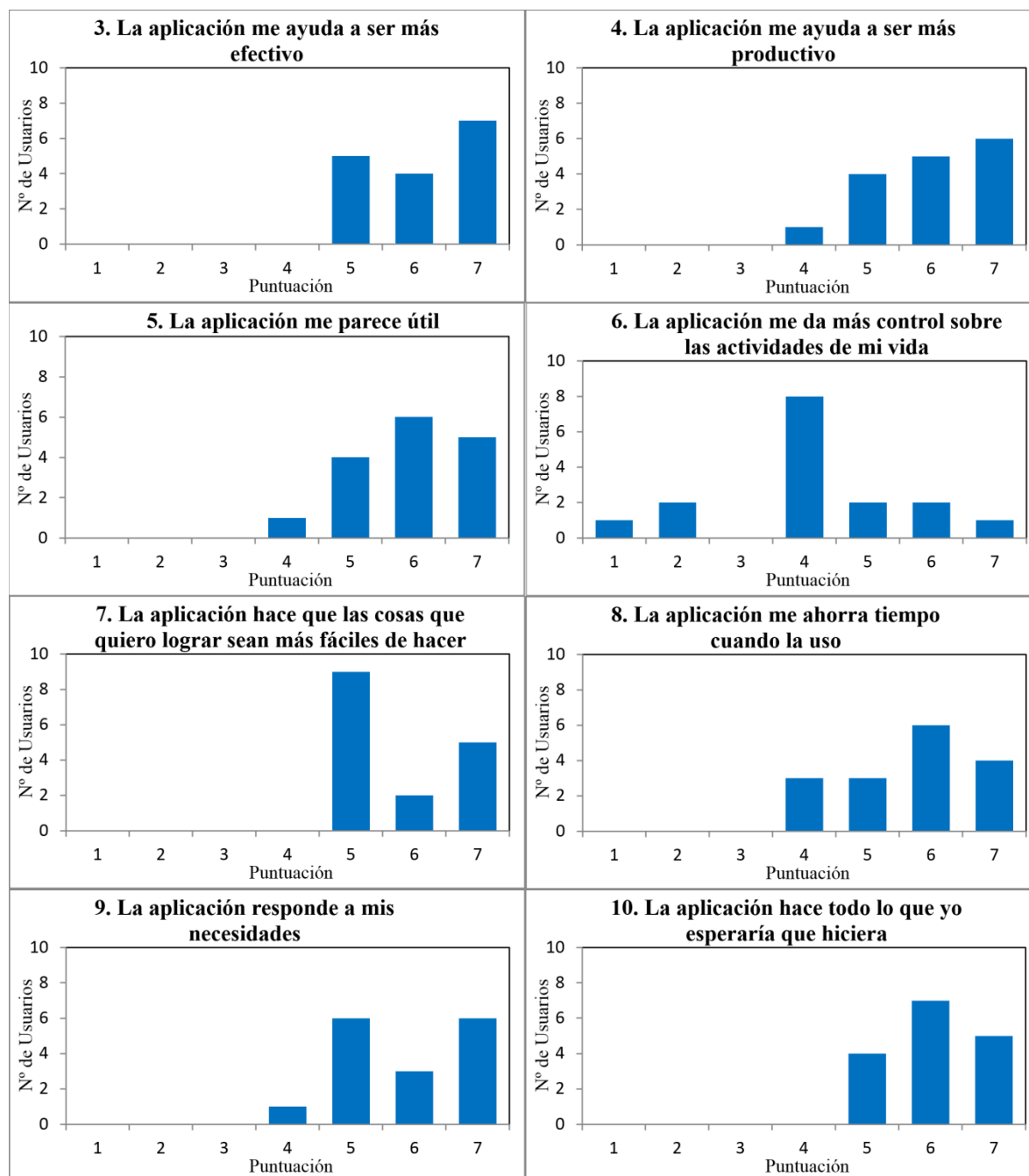
#### a) Utilidad

En este apartado, se presentan los resultados obtenidos en la medición de la dimensión *Utilidad* para la herramienta Scrum-UIA-MAT. La Figura 88 muestra ocho gráficos, correspondientes a las ocho preguntas de esta dimensión (de la 3 a la 10 del cuestionario). En cada gráfico, aparece el número de usuarios (frecuencia) que contestaron a cada pregunta en el eje Y, frente a la puntuación (de 1 a 7) en el eje X que los usuarios otorgaron en cada caso.

En términos generales, se puede indicar que la herramienta Scrum-UIA-MAT obtiene puntuaciones altas en cuanto a la utilidad de la herramienta ( $M=5,7$ ,  $SD=1,18$ ). De hecho, sólo la pregunta 6 (“La aplicación me da más control sobre las actividades de mi vida”) presenta puntuaciones inferiores a 4, pero obtiene una favorable puntuación media de 4,3



(SD=2,38). La desviación estándar de la pregunta 6 es la más alta entre todas las preguntas de esta dimensión. Los resultados dispersos de la pregunta 6 se podrían deber a que usuarios interpretaran que el control de las actividades de sus vidas no está relacionado con las funcionalidades y las características que ofrece la herramienta Scrum-UIA-MAT. Sin embargo, todas las preguntas presentan puntuaciones medias mayores a 4, lo que permite concluir que la utilidad alcanzada es positiva en la totalidad de los casos.



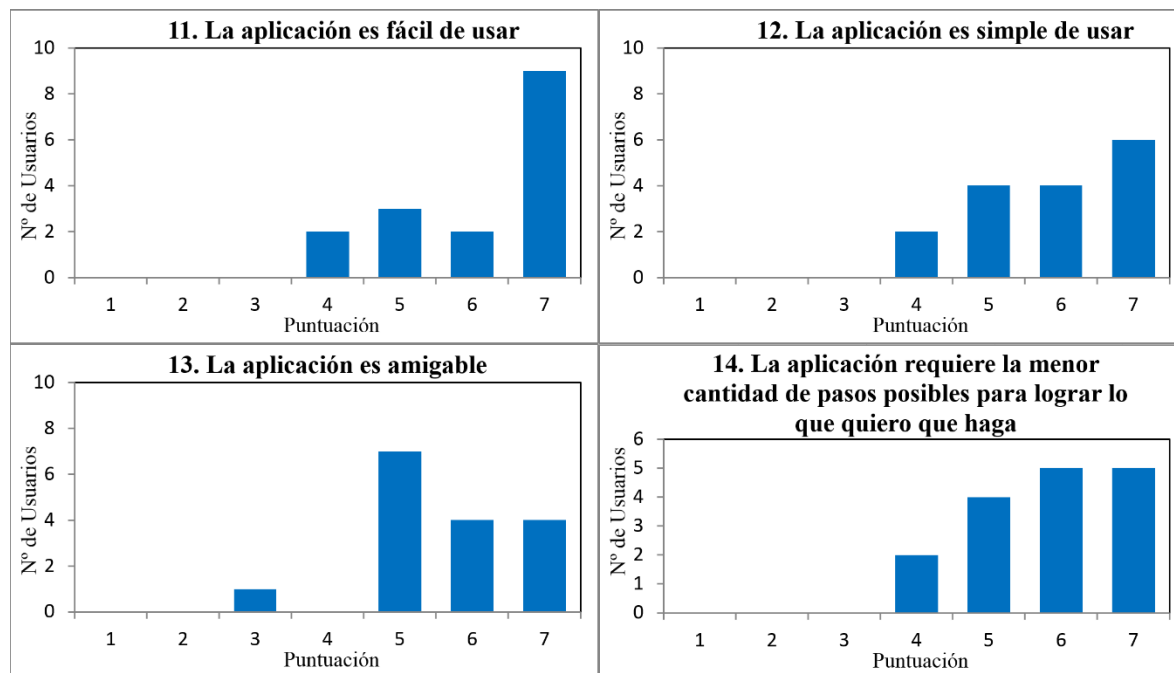
**Figura 88:** Gráficos correspondientes a las ocho preguntas (de 3 a 10) de la dimensión *Utilidad*, con el número de usuarios (frecuencia) que contestaron a cada pregunta (eje y), frente a la puntuación (de 1 a 7) entregada por los usuarios (eje x).

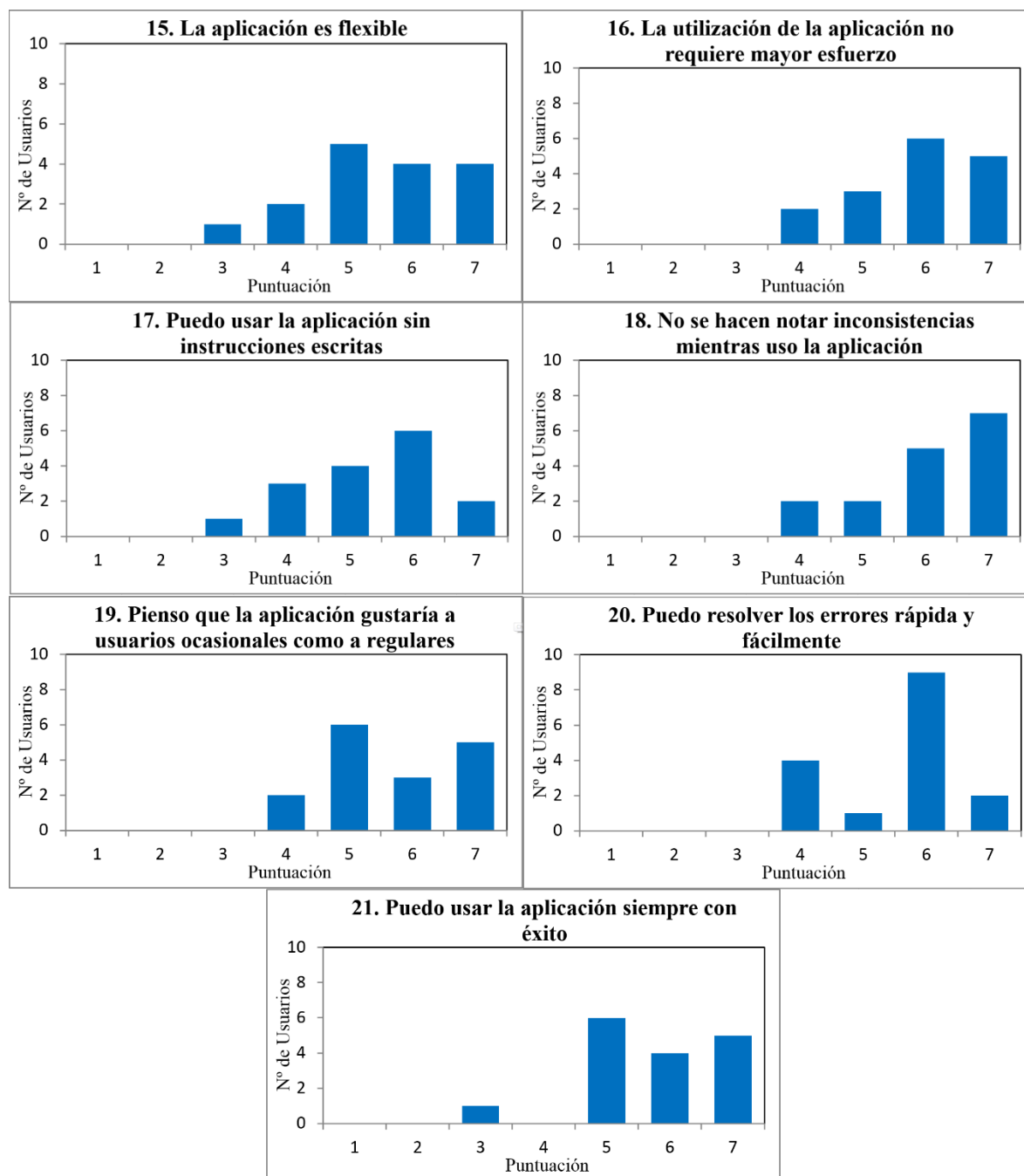
Como se puede observar en la Figura 88, destacan las altas puntuaciones de las preguntas 3 (“La aplicación me ayuda a ser más efectivo”), 4 (“La aplicación me ayuda a ser más productivo”) y 10 (“La aplicación hace todo lo que yo esperaría que hiciera”), que obtienen puntuaciones medias de 6,13 (SD=0,89), 6 (SD=0,97) y 6,06 (SD=0,77), respectivamente. Asimismo, la pregunta 9 (“La aplicación responde a mis necesidades”) destaca por su concentración del 93% de los usuarios con puntuaciones entre 5 y 7 (M=5,88, SD=1,02). Estos resultados se podrían deber a que los usuarios completaron las tareas propuestas sin manifestar la carencia de alguna funcionalidad que les impidiera finalizarlas.

La pregunta 5 (“La aplicación me parece útil”) destaca por su concentración del 68% de los usuarios con puntuaciones entre 6 y 7 (M=5,94, SD=0,93). Esto se podría explicar a que los usuarios consideraran que la herramienta es conveniente y pertinente en los contextos para gestionar proyectos de informática.

### b) Facilidad de Uso

En este apartado, se presentan los resultados obtenidos en la medición de la dimensión *Facilidad de Uso* para la herramienta Scrum-UIA-MAT. La Figura 89 muestra 11 gráficos correspondientes a las 11 preguntas de esta dimensión (del 11 al 21 del cuestionario). En cada gráfico, aparece el número de usuarios (frecuencia) que contestaron a cada pregunta en el eje Y, frente a la puntuación (de 1 a 7) en el eje X que los usuarios otorgaron en cada caso.





**Figura 89:** Gráficos correspondientes a las 11 preguntas (de 11 a 21) de la dimensión *Facilidad de Uso*, con el número de usuarios (frecuencia) que contestaron a cada pregunta (eje y), frente a la puntuación (de 1 a 7) entregada por los usuarios (eje x).

En términos generales, se puede indicar que la herramienta obtiene puntuaciones altas respecto a sus características de utilidad ( $M=5,74$ ,  $SD=1,09$ ). Solamente en 4 de las 11 preguntas (13, 15, 17 y 21) se obtienen puntuaciones inferiores a 4. No obstante, estas preguntas obtienen puntuaciones medias sobre 4, es decir, 5,63 ( $SD=1,09$ ), 5,50 ( $SD=1,21$ ), 5,31 ( $SD=1,14$ ) y 5,75 ( $SD=1,13$ ), respectivamente. Asimismo, se puede destacar que la

pregunta 11 (“La aplicación es fácil de usar”) obtiene una puntuación de 7 por parte del 56% de los usuarios ( $M=6,3$ ,  $SD=1,15$ ). Lo anterior permite concluir que el grado de facilidad de uso alcanzado es positivo en la mayoría de los casos.

Las preguntas número 11 (“La aplicación es fácil de usar”), 12 (“La aplicación es simple de usar”), 16 (“La utilización de la aplicación no requiere mayor esfuerzo”) y 18 (“No se hacen notar inconsistencias mientras uso la aplicación”) obtienen las puntuaciones medias más altas, de 6,13 ( $SD=1,15$ ), 5,88 ( $SD=1,09$ ), 5,88 ( $SD=1,02$ ) y 6,06 ( $SD=1,06$ ), respectivamente. Estos resultados permiten indicar que los usuarios consideran que las funcionalidades y las características de la herramienta tienen niveles bajos de dificultad y esfuerzo y que, a la vez, no presentan inconsistencias en su uso.

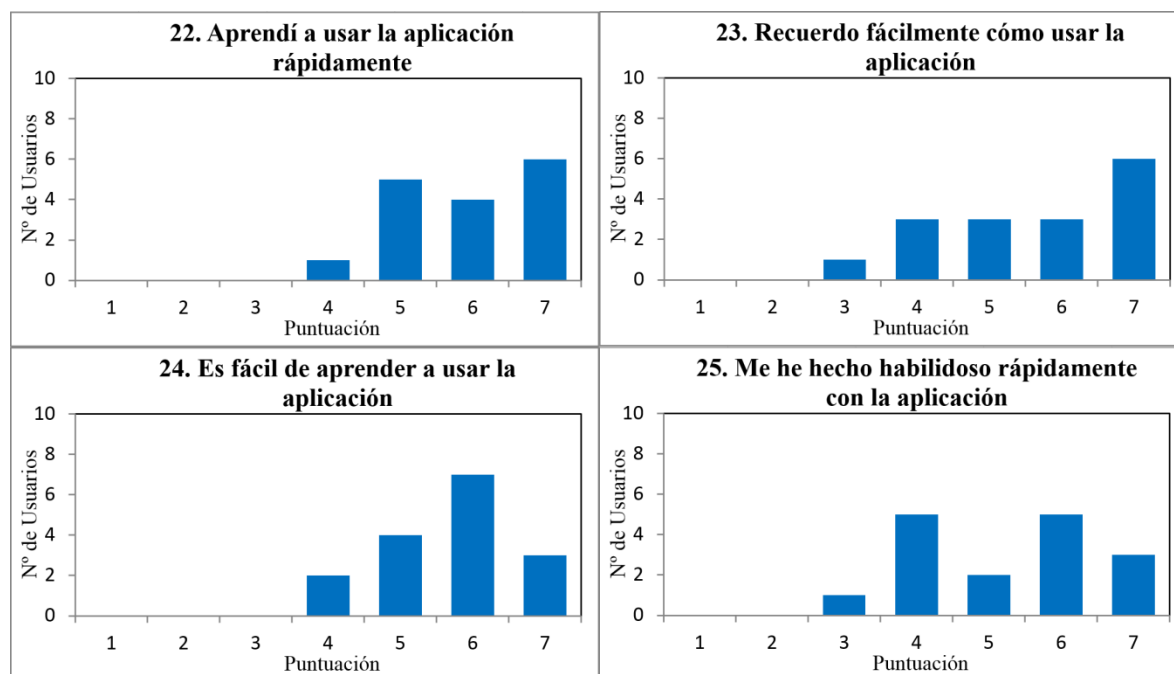
Las preguntas número 14 (“La aplicación requiere la menor cantidad de pasos posibles para lograr lo que quiero que haga”) y 19 (“Pienso que la aplicación gustaría a usuarios ocasionales como a regulares”) obtienen puntuaciones a partir de 5, por parte del 87% de los usuarios. Estas preguntas obtienen puntuaciones medias de 5,81 ( $SD=1,05$ ) y 5,69 ( $SD=1,08$ ), respectivamente.

### c) **Facilidad de Aprendizaje**

En este apartado, se presentan los resultados obtenidos en la medición de la dimensión *Facilidad de Aprendizaje* para la herramienta Scrum-UIA-MAT. La Figura 90 muestra cuatro gráficos correspondientes a las cuatro preguntas de esta dimensión (números 22 al 25 del cuestionario). En cada gráfico, aparece el número de usuarios (frecuencia) que contestaron a cada pregunta en el eje Y, frente a la puntuación (de 1 a 7) en el eje X que los usuarios otorgaron en cada caso.

Nuevamente destaca, al igual que para las dimensiones descritas anteriormente, la alta puntuación obtenida en todas las preguntas ( $M=5,63$ ,  $SD=1,16$ ). Esto permite concluir que el grado de facilidad de aprendizaje alcanzado es positivo en todos los casos.

La pregunta 22 (“Aprendí a usar la aplicación rápidamente”) obtiene la puntuación media más alta, de 5,94 ( $SD=1,00$ ), y puntuaciones a partir de 5 por parte del 93% de los usuarios. Asimismo, la pregunta 24 (“Es fácil de aprender a usar la aplicación”) obtiene la segunda puntuación media más alta, de 5,69 ( $SD=0,9$ ), y puntuaciones a partir de 5 por parte del 87% de los usuarios. Esto se explica, en parte, porque durante la introducción de la evaluación de la herramienta los usuarios solamente requirieron repeticiones de las explicaciones en los temas referentes a los conceptos y la nomenclatura utilizada en las metodologías ágiles, lo que generó que los usuarios aprendieran y recordaran la herramienta de forma fácil y rápida.

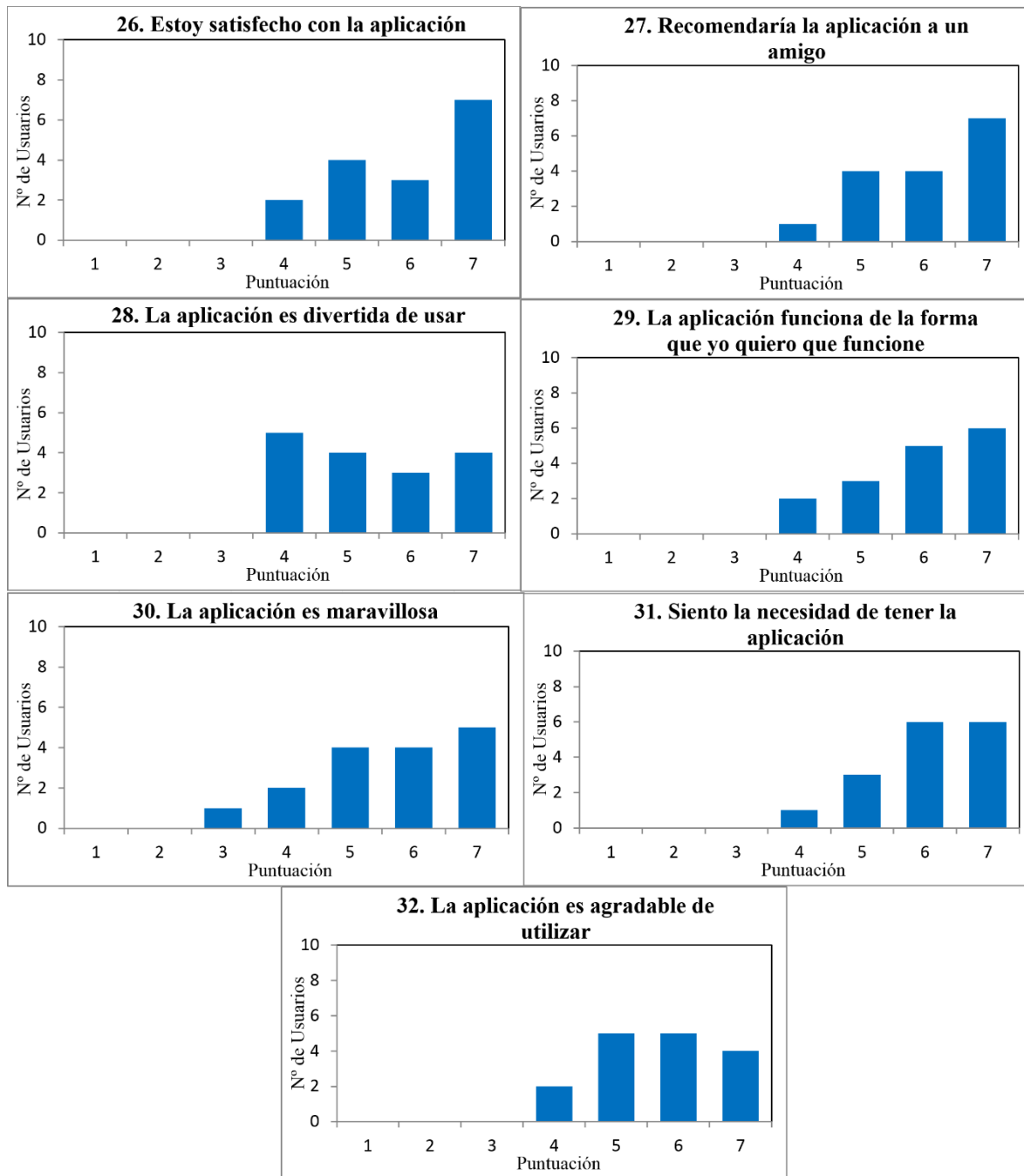


**Figura 90:** Gráficos correspondientes a las cuatro preguntas (de 22 a 25) de la dimensión *Facilidad de Aprendizaje*, con el número de usuarios (frecuencia) que contestaron a cada pregunta (eje y), frente a la puntuación (de 1 a 7) entregada por los usuarios (eje x).

Solamente las preguntas 23 (“Recuerdo fácilmente cómo usar la aplicación”) y 25 (“Me he hecho habilidoso rápidamente con la aplicación”) obtienen puntuaciones inferiores a 4. No obstante, dichas preguntas obtienen puntuaciones medias positivas de 5,63 (SD=1,36) y 5,25 (SD=1,29), respectivamente. Asimismo, las preguntas 23 y 25 obtienen puntuaciones a partir de 5 por parte del 75% y 62% de los usuarios, respectivamente. Los resultados anteriores permiten indicar que el grado de facilidad para recordar y aprender cómo utilizar la herramienta es positivo.

#### d) Satisfacción

En este apartado, se presentan los resultados obtenidos en la medición de la dimensión *Satisfacción* para la herramienta Scrum-UIA-MAT. La Figura 91 muestra siete gráficos correspondientes a las siete preguntas de esta dimensión (números 26 al 32 del cuestionario). En cada gráfico, aparece el número de usuarios (frecuencia) que contestaron a cada pregunta en el eje Y, frente a la puntuación (de 1 a 7) en el eje X que los usuarios otorgaron en cada caso.



**Figura 91:** Gráficos correspondientes a las siete preguntas (de 26 a 32) de la dimensión *Satisfacción*, con el número de usuarios (frecuencia) que contestaron a cada pregunta (eje y), frente a la puntuación (de 1 a 7) entregada por los usuarios (eje x).

Nuevamente destaca la alta puntuación obtenida en todas las preguntas ( $M=5,81$ ,  $SD=1,04$ ). Solamente la pregunta 30 (“La aplicación es maravillosa”) obtuvo una puntuación menor a 4. No obstante, esta pregunta obtuvo una puntuación media de 5,63 ( $SD=1,58$ ) (ésta es la desviación estándar más alta de las preguntas de esta dimensión) y una puntuación a partir

de 5 por parte del 81% de los usuarios. Por lo tanto, los resultados anteriores permiten indicar que el grado de satisfacción alcanzado es positivo en todos los casos.

Las preguntas número 27 (“Recomendaría la aplicación a un amigo”) y 31 (“Siento la necesidad de tener la aplicación”) obtuvieron las puntuaciones medias más altas, con una media de 6,06 en ambos casos ( $SD=1$  y  $SD=0,86$ , respectivamente). Además, ambas preguntas se puntuaron a partir de un 5 por más del 93% de los usuarios. Los resultados anteriores permiten indicar que los usuarios tienen un alto nivel de satisfacción en el uso de la herramienta, considerando que la herramienta es recomendable y que la necesitan tener.

Las preguntas 26 (“Estoy satisfecho con la aplicación”) y 29 (“La aplicación funciona de la forma que yo quiero que funcione”) destacan por su concentración de más del 87% de los usuarios con puntuaciones entre 5 y 7. Aquí las preguntas 26 y 29 obtuvieron una puntuación media de 5,94 en ambos casos ( $SD=1,26$  y  $SD=1,45$ , respectivamente). Esto permite afirmar que la herramienta cuenta con un nivel alto de aprobación, ya que sus funcionalidades trabajan de manera esperada por los usuarios.

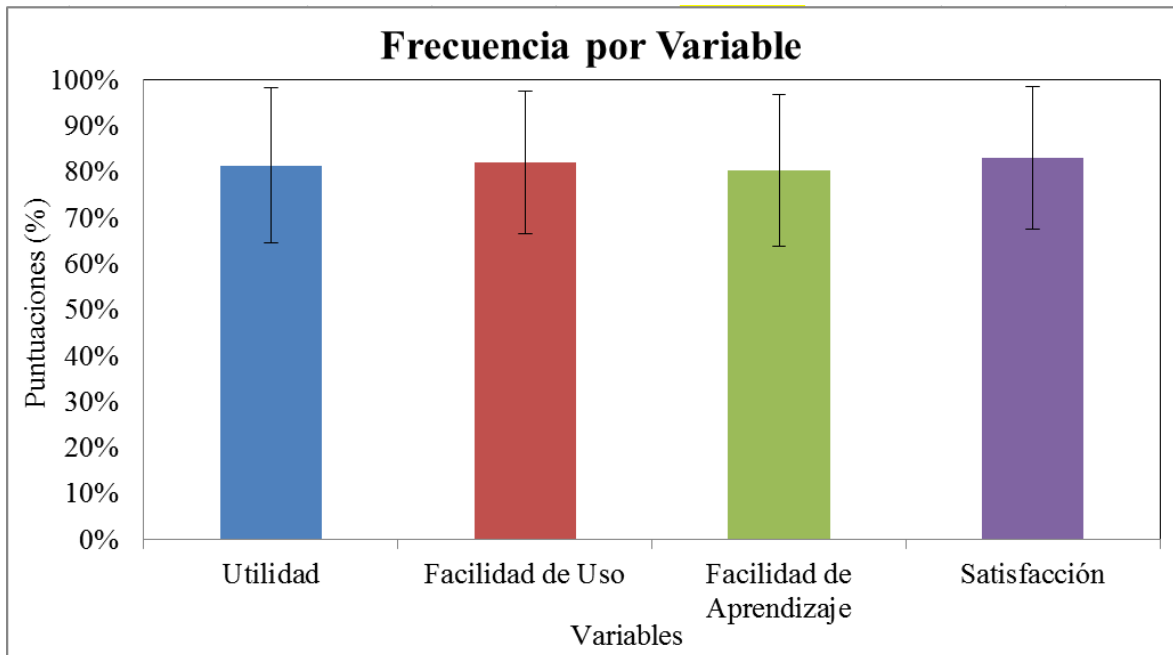
### **5.3.3.3. Fiabilidad de la Evaluación**

Tal y como se mencionó anteriormente en la sección 5.2.3.3, la fiabilidad se puede considerar como una medida de confianza de los resultados, emitidos por una determinada evaluación. Por lo tanto, la fiabilidad de la evaluación realizada a los 16 usuarios se ha medido a través del indicador Alfa de Cronbach. Este indicador se ha calculado para las 30 preguntas cerradas del cuestionario. El resultado muestra un valor de fiabilidad del 96,43% ( $\alpha=0,96$ ), lo que permite indicar que el cuestionario presenta un nivel de fiabilidad excelente, superando el umbral de aceptación en un 26,43%.

### **5.3.3.4. Análisis Multivariable**

En las secciones anteriores, sólo se han indicado los resultados del cuestionario, agrupados de acuerdo a las diferentes dimensiones medidas, sin incluir una evaluación orientada a establecer comparaciones entre las dimensiones de forma conjunta. Por lo tanto, en esta sección, se presenta un análisis multivariable, que tiene como objetivo analizar, de forma simultánea, las cuatro dimensiones medidas (*Utilidad, Facilidad de Uso, Facilidad de Aprendizaje y Satisfacción*).

Para llevar a cabo este análisis, primero se realizó una serie de cálculos con los datos correspondientes a cada una de las dimensiones medidas, tales como la puntuación promedio y la desviación estándar de cada una de las dimensiones y del total de la muestra. La Figura 92 presenta el resultado de los cálculos de las cuatro dimensiones mencionadas anteriormente. Este gráfico muestra los distintos valores medios de las dimensiones normalizadas (entre 0 y 100%), una barra correspondiente ( $\pm \sigma$ ) a la dispersión de cada dimensión (desviación estándar), así como la media general (línea de puntos horizontal).



**Figura 92:** Usabilidad de Scrum-UIA-MAT. Puntuación media para cada dimensión que se mide en valores normalizados entre 0 y 100%, incluyendo las barras de error ( $\pm \sigma$ ) y una línea de puntos que representa la media.

Como se puede observar en la Figura 92, la dimensión *Satisfacción* es la que alcanza la puntuación promedio más alta de toda la muestra, con una media de 83,03% (SD=15,52%). Por otro lado, la dimensión *Facilidad de Uso* obtiene el segundo promedio más alto de las puntuaciones medidas, con un promedio de 82,06% (SD=15,56%). Ésta es seguida por la dimensión *Utilidad*, que obtiene una puntuación promedio de 81,36% (SD=16,87%). Finalmente, la dimensión *Facilidad de Aprendizaje* fue la que obtuvo el promedio de puntuaciones más bajo, de 80,35% (SD=16,59%), que corresponde también a la dispersión más alta entre las dimensiones medidas.

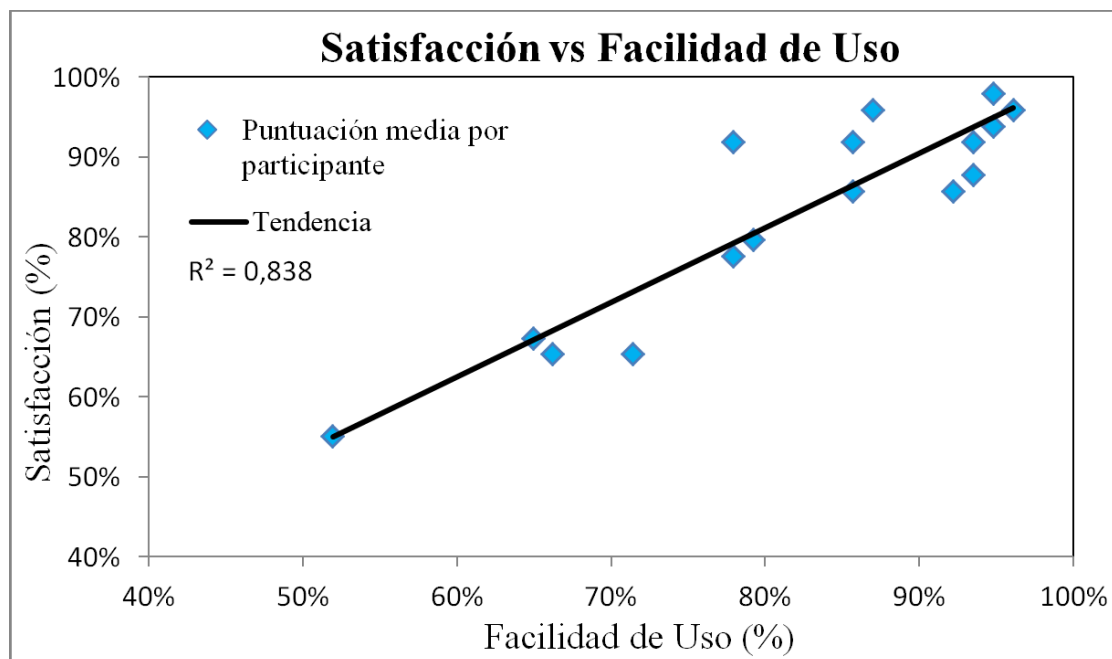
En términos generales, se puede indicar que el promedio de las puntuaciones de toda la muestra es de 81,7% (SD=16,14%). Cabe destacar que todas las dimensiones obtienen puntuaciones superiores a 80%, lo que se considera un buen indicador de la usabilidad general de la herramienta Scrum-UIA-MAT, respecto a la percepción de los usuarios.

Debido a que los resultados de las medias de las cuatro dimensiones están próximos entre sí, se estudió la posible correlación de los valores medios. Así pues, se efectuó un cálculo para medir el coeficiente de correlación entre las dimensiones medidas. Este análisis tiene como objetivo medir el grado de intensidad lineal de vinculación de las dimensiones. Para realizar este cálculo, primero se confeccionó una matriz de correlación para estudiar la correlación de todas las dimensiones y así poder elegir las dimensiones que estaban más correlacionadas (considerando un umbral superior al 70% del coeficiente de correlación). A continuación, se generaron gráficos de dispersión de las puntuaciones promedio de cada una de las dimensiones seleccionadas, confrontando las puntuaciones unas con otras, y



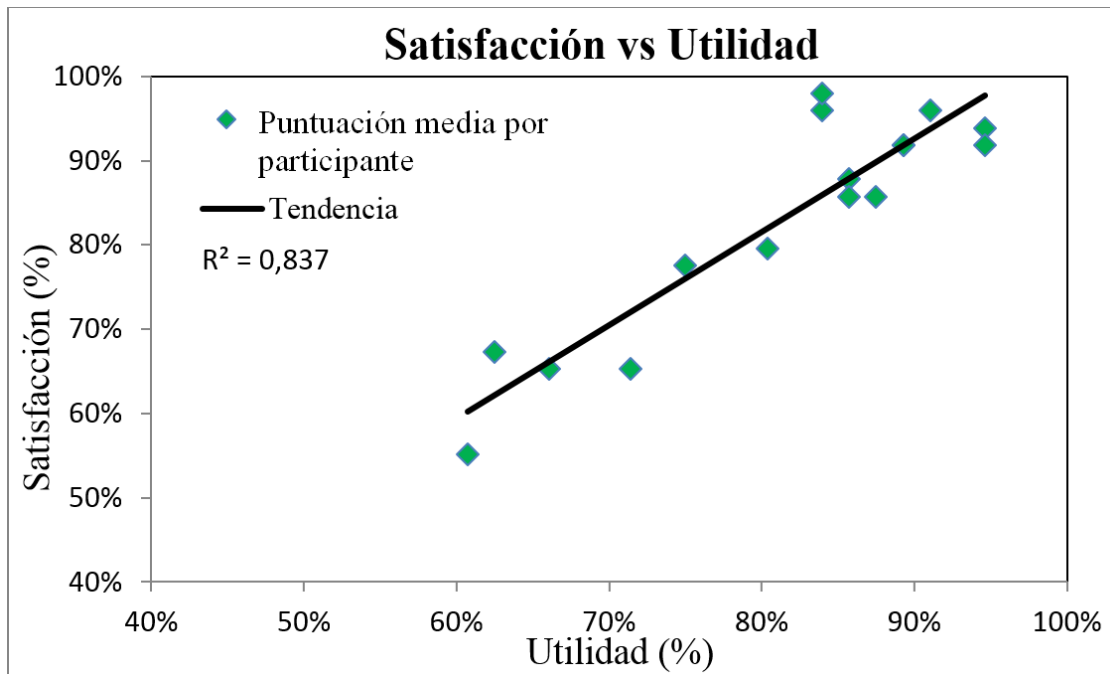
calculando el coeficiente de correlación lineal ( $R^2$ ). Este coeficiente se sitúa entre 0 y 1, teniendo una mayor relación lineal entre las dimensiones estudiadas al acercarse al 1.

La correlación más alta encontrada fue entre las dimensiones *Satisfacción* y *Facilidad de Uso*, con un coeficiente de correlación lineal en sentido positivo de  $R^2=0,838$ . Se ha elaborado la Figura 93 para presentar la correlación entre estas dimensiones. Dicha correlación permite señalar que los usuarios manifiestan mayor satisfacción en la utilización de la herramienta Scrum-UIA-MAT, en la medida que ésta provee una mayor facilidad de uso.



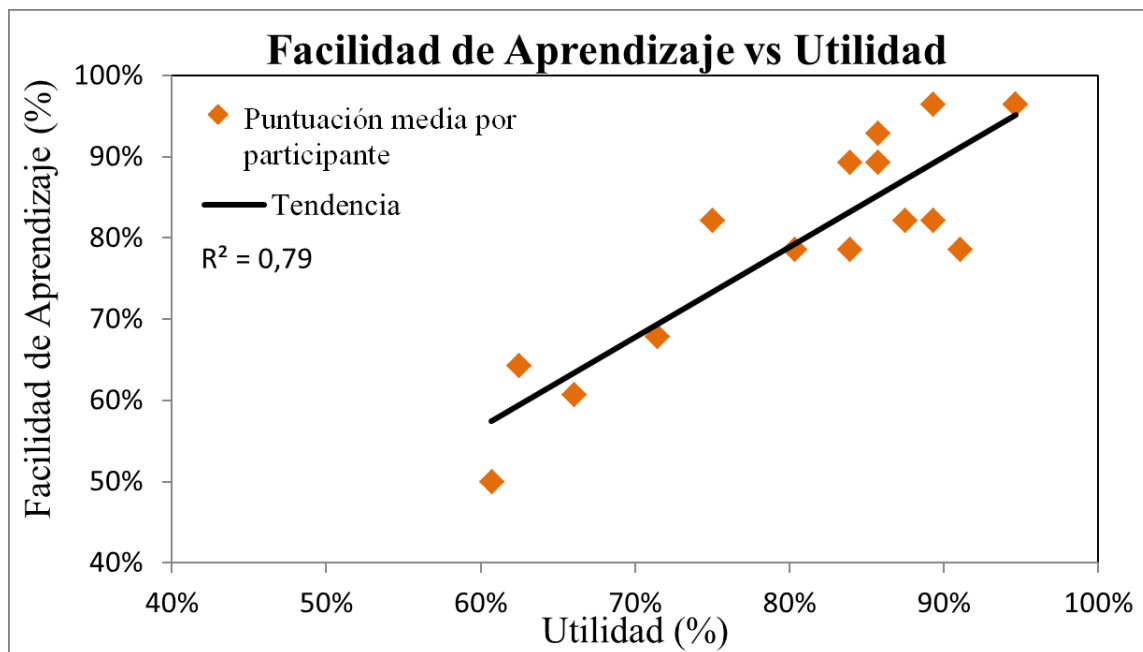
**Figura 93:** Correlación entre las dimensiones *Satisfacción* y *Facilidad de Uso*.

La segunda correlación más alta se presenta en la Figura 94 que corresponde a las dimensiones *Satisfacción* y *Utilidad*. Esta correlación obtiene un coeficiente lineal de  $R^2=0,837$ , lo que demuestra que existe una correlación en sentido positivo entre estas dimensiones. Por lo tanto, se puede señalar que los usuarios manifiestan mayor satisfacción en la utilización de la herramienta Scrum-UIA-MAT, en la medida que ésta cumple con las expectativas de utilidad.



**Figura 94:** Correlación entre las dimensiones *Satisfacción* y *Utilidad*.

Finalmente, la tercera correlación más alta se presenta en la Figura 95 que corresponde a las dimensiones *Facilidad de Aprendizaje* y *Utilidad*. Esta correlación obtiene un coeficiente de correlación lineal de  $R^2=0,79$ , lo que demuestra que existe una correlación en sentido positivo entre estas dimensiones. Por lo tanto, esto permite señalar que los usuarios manifiestan una mayor facilidad en el aprendizaje en la herramienta Scrum-UIA-MAT, en la medida que ésta cumple con las expectativas de utilidad.



**Figura 95:** Correlación entre las dimensiones *Facilidad de Aprendizaje* y *Utilidad*.

### 5.3.4. Discusión de los Resultados

En esta sección se efectúa una discusión de los resultados, obtenidos en la sesión experimental con los 16 usuarios y en el cuestionario de usabilidad.

En cuanto al grado de eficacia de las tareas del experimento, los resultados muestran que en todos los casos los usuarios finalizaron las tareas correctamente, alcanzado un 100% de eficacia. Se destaca que aun cuando 5 de las 12 tareas requirieron algún tipo de asistencia, el porcentaje de asistencia requerida es bajo en todos los casos, con un promedio de 3,16% ( $SD=4,51\%$ ).

Las dudas y consultas realizadas por los usuarios proveen información importante para mejorar algunos aspectos de la herramienta. Por ejemplo, la opción para asignar un requisito a un Sprint no entregaba retroalimentación a los usuarios en los casos donde se requería seleccionar el requisito para completar la tarea. Asimismo, la imagen proporcionada para listar las tareas de desarrollo requería un texto alternativo de ayuda, con el fin de guiar a los usuarios. Finalmente, las opciones ofrecidas para ordenar los requisitos no estaban correctamente separadas, con el fin de identificarlas con mayor claridad.

Por otro lado, en cuanto a la eficiencia, al examinar los tiempos medios utilizados en cada tarea, se puede indicar que son razonables, debido a que 10 de las 12 tareas tienen un promedio de 2 minutos de realización (a excepción de las tareas T4 y T6, que son las que más tiempo exigieron a los usuarios).

Cabe destacar que aun cuando la tarea T6 requirió una mayor cantidad de tiempo a los usuarios, con una duración promedio de 9 minutos y 57 segundos ( $SD=2$  minutos y 35 segundos), esta tarea se llevó a cabo sin que los usuarios requirieran asistencia. Asimismo, las tareas que más tiempo demandaron tenían en común el hecho de que requerían digitar una mayor cantidad de información. Estos resultados permiten indicar que la herramienta no presentaba dificultades arquitectónicas ni de diseño, que influyeran en el tiempo de realización de las tareas, ya que está sujeta a la cantidad de información que necesitaban ingresar los usuarios en cada caso.

Respecto a los navegadores Web utilizados por los usuarios, éstos no incidieron en los tiempos utilizados para completar las tareas, ni tampoco fueron causantes de las asistencias requeridas en la completación de las tareas. Los usuarios que participaron en la evaluación utilizaron diferentes tipos de navegadores Web, y se pudo constatar que la herramienta Scrum-UIA-MAT se ejecutó sin generar inconvenientes que impidieran la realización correcta de las tareas por parte de los usuarios.

En cuanto a la satisfacción, los resultados del cuestionario de usabilidad muestran unos valores altos para las dimensiones *Satisfacción* (83,03%) y *Facilidad de Uso* (82,06%). Destaca el resultado de la pregunta número 11 (“La aplicación es fácil de usar”) que consigue una puntuación media de 90% ( $SD=16,42\%$ ) y una puntuación del 100% por parte del 56% de los usuarios. Estos resultados de las dimensiones, junto con el grado de eficacia

logrado en las tareas del experimento, permiten confirmar que la herramienta provee cualidades y un modelo arquitectónico que son fáciles de manejar, y que han generado satisfacción entre los usuarios.

En cuanto a las preguntas abiertas del cuestionario (“Listar los aspectos más negativos” y “Listar los aspectos más positivos”), éstas entregaron datos relevantes respecto a los aspectos de mejora para la herramienta Scrum-UIA-MAT. En la pregunta “Listar los aspectos más negativos”, los usuarios aportaron datos valiosos a través de los siguientes comentarios: *“algunos módulos no entregan feedback al guardar los datos”*, *“falta mantener el foco en los botones principales de cada formulario”*, *“falta resaltar el nombre del proyecto”*, *“opciones para editar tareas deberían aparecer en el listado de las tareas”*, *“la opción para agregar elementos debería estar junto con la opción de los aspectos”* y *“algunas imágenes no eran posibles de visualizar de manera completa”*. Los comentarios y observaciones anteriores aportan información de entrada importante para continuar mejorando los diferentes aspectos de la herramienta Scrum-UIA-MAT.

Por otro lado, los aspectos positivos, destacados por los usuarios, han sido los siguientes: *“diseño sencillo”*, *“estructura fácil de recordar”*, *“no arrojó errores”*, *“responde de manera rápida”*, *“ahorra tiempo”*, entre otros. Tales comentarios permiten verificar la evaluación positiva, conseguida por la herramienta en las preguntas cerradas.

Asimismo, los resultados obtenidos proporcionan evidencia suficiente para responder las preguntas de la investigación **PI<sub>21</sub>**, **PI<sub>22</sub>** y **PI<sub>23</sub>**, indicadas anteriormente:

- **PI<sub>21</sub>**: ¿Cuál es la eficacia alcanzada por los usuarios en la realización de tareas de gestión de un proyecto utilizando Scrum-UIA-MAT?

Los resultados muestran que en todos los casos los usuarios finalizaron las tareas correctamente, por lo tanto, se puede indicar que se ha alcanzado un 100% de eficacia.

- **PI<sub>22</sub>**: ¿Cuál es la eficiencia alcanzada por los usuarios en la realización de tareas de gestión de un proyecto utilizando Scrum-UIA-MAT?

La realización completa de las tareas tuvo una duración promedio de 25 minutos y 49 segundos (SD=8 minutos y 4 segundos). La mayoría de los tiempos medios utilizados en cada tarea por los usuarios rondaba los 2 minutos. También destacan los intervalos de confianza de los tiempos de realización de estas tareas, que tienen valores inferiores a un minuto, con excepción a la tarea T6, que tiene un intervalo de confianza de 1 minuto y 22 segundos. Estos resultados se consideran aceptables para las operaciones solicitadas a los usuarios, y proveen una ventaja para apoyar la gestión de proyectos software.

- **PI<sub>23</sub>**: ¿Cuál es la percepción de usabilidad de los usuarios con respecto a Scrum-UIA-MAT?

La percepción de usabilidad de los 16 Ingenieros del Software con respecto a la herramienta Scrum-UIA-MAT es positiva en todos los casos. Los resultados del

cuestionario de usabilidad obtiene un promedio de toda la muestra de 83,03% (SD=15,52%), siendo un buen indicador de la percepción de usabilidad de los usuarios con respecto a la herramienta propuesta.

Finalmente, todos los resultados anteriores permiten corroborar la última hipótesis de partida **H2.4**, al concluir que la herramienta Scrum-UIA-MAT es fácil de usar y de aprender por parte de sus usuarios, y que asimismo permite acceder y recopilar la información de las distintas etapas de la gestión de proyectos con Scrum-UIA, facilitando gestionar colectivamente los requisitos, de acuerdo a las prioridades del proyecto, y revisar el cumplimiento de las tareas de desarrollo, considerando aspectos relacionados con los usuarios finales.

### 5.4. Recapitulación

A lo largo de este capítulo, se han presentado diferentes evaluaciones para examinar las características y las funcionalidades de los componentes propuestos para fomentar el desarrollo ágil y centrado en el usuario de la metodología Scrum-UIA. Estos componentes corresponden al método de priorización QMPSR, la herramienta CASE InterArch y la herramienta Web Scrum-UIA-MAT.

Para la evaluación de QMPSR, se ha creado un marco experimental que permite realizar evaluaciones comparativas con otros seis métodos de priorización, y llevar a cabo nueve experimentos para medir las colisiones con diferentes conjuntos de requisitos de entrada. Los resultados de estos experimentos permiten indicar que QMPSR puede adaptarse a entornos complejos y dinámicos, presentando un buen comportamiento en términos de escalabilidad. Por lo tanto, es posible afirmar que QMPSR permite que la metodología Scrum-UIA dirija el proceso de priorización de los requisitos a través de los aspectos y los elementos relevantes de los proyectos, generando una clasificación final que reduce las colisiones de requisitos.

En cuanto a la evaluación de la herramienta CASE InterArch, se ha realizado un diseño cuasi experimental para medir su usabilidad con 12 Arquitectos de la Información. Los resultados de esta evaluación muestran valoraciones más que aceptables, destacando la alta valoración obtenida para las dimensiones de *Facilidad de Aprendizaje* y *Facilidad de Uso*. Por lo tanto, es posible afirmar que InterArch es fácil de usar y de aprender por el Arquitecto de la Información, y permite promover el desarrollo en Scrum-UIA de manera incremental, mediante la representación inicial de los modelos de contenido, creados dentro del dominio del problema por parte del Arquitecto de la Información, y que evolucionan a fases más cercanas al dominio de la solución.

Respecto a la evaluación de la herramienta Scrum-UIA-MAT, se ha elaborado un diseño cuasi experimental para medir su usabilidad con 16 Ingenieros del Software. Estos resultados muestran que la percepción de usabilidad de los usuarios es positiva en todos los casos, indicando unos valores altos para las dimensiones de *Satisfacción* y *Facilidad de*

*Uso*. Por lo tanto, es posible afirmar que la herramienta Scrum-UIA-MAT es fácil de usar y de aprender por parte de sus usuarios potenciales, y suministra un sistema eficiente para la gestión ágil de proyectos, basado en la Scrum-UIA.

En cuanto a las evaluaciones de la usabilidad de las herramienta InterArch y Scrum-UIA-MAT, ésta última obtiene los mejores resultados para las dimensiones *Facilidad de Uso* ( $M=82,06\%$ ,  $SD=15,56\%$ ), *Utilidad* ( $M=81,36\%$ ,  $SD=16,87\%$ ) y *Facilidad de Aprendizaje* ( $80,35\%$ ,  $SD=16,59\%$ ). Solamente en la dimensión *Facilidad de Aprendizaje* la herramienta InterArch ( $M=83,05\%$ ,  $SD=12,3\%$ ) obtiene resultados mejores que Scrum-UIA-MAT ( $M=80,35\%$ ,  $SD=16,59\%$ ). Esto es debido a que, posiblemente, la herramienta Scrum-UIA-MAT exige manipular más módulos y funcionalidades por parte del usuario, para llevar a cabo la gestión de cada proyecto, suponiendo un aumento en la exigencia para aprender, de manera eficiente, cada uno de ellos.

Finalmente, estos resultados también permiten afirmar, como se indicó al inicio de esta tesis doctoral, que tanto la metodología Scrum-UIA como sus componentes propuestos (QMPSR, InterArch y Scrum-UIA-MAT) son una alternativa competitiva para los equipos de trabajo de informática que necesitan adaptarse a situaciones de alta complejidad y cambio, así como involucrar a los usuarios finales en todo el ciclo de vida del proceso de desarrollo de software, debido a que se provee un marco de trabajo flexible que posibilita el logro de los objetivos propuestos y la obtención de sistemas fáciles de utilizar y de aprender, minimizando el coste, el tiempo y el esfuerzo de su construcción.

# Capítulo 6: Conclusiones y Trabajo Futuro

---

Una vez descrita, verificada y evaluada la solución propuesta, en este capítulo se describen las conclusiones más importantes de la investigación realizada y de las actividades que se consideran como líneas futuras para continuar la tesis doctoral. En primer lugar, en la sección 6.1, se presentan las conclusiones principales de la investigación. Posteriormente, en la sección 6.2, se describen las lecciones aprendidas durante la realización de esta tesis doctoral. Finalmente, en la sección 6.3, se detallan las oportunidades para continuar la presente investigación.

## 6.1. Principales Conclusiones de la Tesis Doctoral

Hoy en día, la AI es un término de relevancia que cobra sentido en el desarrollo de aplicaciones y de sistemas de información. De hecho, existen diversas propuestas para enfrentar el desafío de integrar la AI de una manera eficiente. Sin embargo, la mayoría de las propuestas actuales tienen dificultades para adaptarse a entornos cambiantes y priorizar dinámicamente los requisitos de usabilidad de contenido. Asimismo, estas propuestas no proveen, de forma automática o semi-automática, información de análisis y diseño para Analistas e Ingenieros del Software, a partir de la información de análisis conceptual del Arquitecto de la Información, que pueda dar continuidad al resto de las fases del ciclo de vida del proyecto. Todos estos inconvenientes dificultan, además, la integración de la AI en un entorno de desarrollo ágil, pues aumentan el tiempo para conseguir incrementos de software que los usuarios finales puedan evaluar y validar.

Con el objetivo de resolver estos inconvenientes, se ha propuesto una integración de la AI dentro de un proceso ágil de desarrollo centrado en el usuario que permite alinear eficientemente las necesidades de los usuarios finales y del negocio con el desarrollo de aplicaciones interactivas.

El análisis bibliográfico, efectuado a través de un estudio sistemático de la literatura, detallado en la sección 2.1, demostró la inexistencia de contribuciones previas que aborden la problemática mencionada y permitió corroborar la hipótesis de partida **H1.1**, así como responder la pregunta de investigación **PI<sub>1</sub>**. Asimismo, se analizaron las diferentes propuestas para la AI, y se comprobó que éstas no permiten integrarla en un modelo de proceso ágil de desarrollo, que considere a los usuarios finales durante todo el ciclo de vida del desarrollo de software. Esto permitió corroborar la hipótesis de partida **H1.2** y responder las preguntas de investigación **PI<sub>2</sub>**, **PI<sub>3</sub>**, **PI<sub>4</sub>** y **PI<sub>5</sub>**.

Continuando con el estudio anterior, se analizaron las metodologías ágiles con el fin de distinguir su viabilidad para incorporar el Diseño Centrado en el Usuario en la integración de la AI. Esto permitió identificar que la metodología ágil Scrum presenta un marco de trabajo ideal para considerar y analizar los requisitos de los usuarios finales, corroborando la hipótesis de partida **H1.3** y respondiendo las preguntas de investigación **PI<sub>7</sub>**, **PI<sub>8</sub>** y **PI<sub>9</sub>**. Seguidamente, con el objetivo de integrar la AI, se examinaron las recomendaciones específicas para incorporar el Diseño Centrado en el Usuario en el proceso ágil de desarrollo, y se logró identificar las directrices principales para diseñar la solución propuesta, permitiendo corroborar la hipótesis de partida **H1.4** y responder las preguntas de investigación **PI<sub>10</sub>** y **PI<sub>11</sub>**. Finalmente, se analizaron los inconvenientes de los métodos de priorización actuales, logrando identificar que éstos no permiten conducir dinámicamente el proceso de priorización, considerando las prioridades de la usabilidad, de la AI y del valor de negocio. Esto permitió corroborar la hipótesis de partida **H1.5** y responder las preguntas de investigación **PI<sub>12</sub>**, **PI<sub>13</sub>** y **PI<sub>14</sub>**.

Así pues, tras una exhaustiva revisión de la literatura y la posterior corroboración de las hipótesis de partida, la aportación principal de esta tesis doctoral consistió en una metodología, llamada Scrum-UIA (**Scrum** driven by **Usable Information Architecture**), que permite integrar la AI a través de un proceso ágil de desarrollo centrado en el usuario. Esta metodología se basa en Scrum y en una combinación de las recomendaciones y prácticas específicas que facilitan integrar el DCU dentro del paradigma ágil, aportando las siguientes características:

- La perspectiva de los usuarios finales se analiza, describe y considera durante el proceso de gestión de la Lista del Producto, dirigida por la investigación contextual.
- La implementación de los requisitos se dirige por las prioridades de la usabilidad, de la AI y del valor de negocio a través de una planificación del Sprint, dirigida por los entregables de la AI. Además, se proporciona un conjunto de técnicas ágiles para integrar la AI, con el fin de apoyar la elaboración de las tareas específicas durante la Ejecución del Sprint.
- El cumplimiento de los criterios de aceptación del usuario final se comprueba en las primeras etapas del Sprint, y los incrementos del producto se evalúan a través de los procesos de inspección y mejora continua, dirigidos por el usuario final.

En la metodología Scrum-UIA, se prescribieron 16 actividades, organizadas en los grupos de Investigación Contextual, Gestión de Requisitos, Planificación del Sprint e Inspección y Mejora Continua. Se prescribieron un total de 29 tareas y 31 productos para llevar a cabo cada actividad, así como las técnicas, los métodos y los roles responsables para su ejecución.

A continuación, se detallan las actividades prescritas en la metodología Scrum-UIA:

- Para el Grupo de Actividades de la Investigación Contextual, la Actividad 1.1, la Actividad 2.1 y la Actividad 1.3 se centraron en la identificación y el análisis de los



usuarios finales, del contenido y del contexto del proyecto. Por su parte, la Actividad 1.4 se centró en la identificación del panorama general del proyecto para generar una visión global del producto, con respecto a las prioridades de la AI, del valor de negocio y de la usabilidad.

- Respecto al Grupo de Actividades de la Gestión de Requisitos, la Actividad 2.1 se dedicó a la identificación de los requisitos del proyecto. Por su parte, la Actividad 2.2 se centró en el análisis de los requisitos, priorizando cada uno de ellos de acuerdo a las preferencias del proyecto (identificadas de manera formal en el Grupo de Actividades de la Investigación Contextual). Junto a ello, la Actividad 2.3 se dedicó a la identificación de una definición de “Terminado” (esto es, un acuerdo del equipo Scrum-UIA para determinar cuándo un requisito se encuentra en condiciones para presentarse al final del Sprint) para los requisitos del Sprint, con el fin de proveer un entendimiento común sobre el alcance y las demandas de los requisitos. Finalmente, la Actividad 2.4 se centró en la identificación de un objetivo inicial para el Sprint.
- En cuanto al Grupo de Actividades de la Planificación del Sprint, la Actividad 3.1 se centró en la determinación de la velocidad y de la disponibilidad de los recursos humanos que tiene el Equipo de Desarrollo para el Sprint. Por su parte, la Actividad 3.2 se dedicó a la determinación del objetivo que se utiliza para guiar la Ejecución del Sprint. Junto a ello, la Actividad 3.3 se enfocó en la identificación de los requisitos que el Equipo de Desarrollo realiza durante el Sprint, y que forman parte de la Lista de Pendientes del Sprint. Además, la Actividad 3.4 se centró en la identificación y la determinación de las tareas específicas de desarrollo que se requieren para implementar los requisitos del Sprint. Finalmente, la Actividad 3.5 se enfocó en la organización de la ejecución de las tareas de desarrollo específicas, identificadas en la Actividad 3.4, para promover el desarrollo incremental a través de los niveles de fidelidad de los entregables de la AI.
- Respecto al último Grupo de Actividades de Inspección y Mejora Continua, la Actividad 4.1 se centró en la revisión del estado de las tareas de desarrollo y de los problemas que tienen los miembros del Equipo de Desarrollo para llevarlas cabo, con el fin de efectuar un seguimiento del cumplimiento de los criterios de aceptación de los usuarios finales. Por su parte, la Actividad 4.2 se concentró en la inspección y la revisión del incremento del producto, generado por el Equipo de Desarrollo durante el Sprint, con la participación de los usuarios finales. Finalmente, la Actividad 4.3 se enfocó en la inspección de las prácticas utilizadas durante el Sprint, con el objetivo de identificar mejoras relacionadas con el siguiente Sprint y los usuarios finales.

Por otro lado, tal y como se indicó en el Grupo de Actividades de la Gestión de Requisitos, en la metodología Scrum-UIA los requisitos se priorizan considerando formalmente las prioridades del proyecto, identificadas en el Grupo de Actividades de la Investigación

Contextual. Para llevar a cabo lo anterior, se propuso un nuevo método de priorización, llamado QMPSR (**Q**ualitative **M**ethod for **P**rioritizing **S**oftware **R**equirements), adaptado a la metodología Scrum-UIA que permite dirigir dinámicamente el proceso de priorización mediante las prioridades de la usabilidad, de la AI y del valor de negocio.

Como se indicó en el Grupo de Actividades de la Planificación del Sprint, la implementación de los requisitos se organiza para promover el desarrollo incremental en Scrum-UIA, de acuerdo a los entregables de la AI. Así pues, se propuso la técnica InterArch-T (**I**nteroperable **I**nformation **A**rchitecture **T**echnique) para promover el desarrollo incremental en Scrum-UIA mediante los modelos de contenido, creados por el Arquitecto de la Información en las fases más iniciales del proyecto, dentro del dominio del problema, y que evolucionan a fases más cercanas al dominio de la solución (diagramas de clases UML para los miembros del Equipo de Desarrollo). Esta técnica se implementó mediante la construcción de una herramienta CASE, llamada InterArch (**I**nteroperable **I**nformation **A**rchitecture), que se puede utilizar para elaborar las tareas relacionadas con la AI.

Por otro lado, la metodología Scrum-UIA, así como su método de priorización (QMPSR), se ha implementado mediante la construcción de una herramienta Web, llamada Scrum-UIA-MAT (**S**crum-**U**IA **M**anagement **T**ool). Dicha herramienta provee un sistema de gestión de proyectos basado en Scrum-UIA, y proporciona distintas funcionalidades para promover el desarrollo ágil y centrado en el usuario de la AI. La utilización de la herramienta Scrum-UIA-MAT permite:

- La definición, organización y recopilación de las prioridades y de los requisitos del proyecto.
- La contribución de mecanismos de control y seguimiento del cumplimiento de los requisitos y de las tareas específicas que el Equipo de Desarrollo debe realizar durante la Ejecución del Sprint.
- La elaboración, configuración y seguimiento de cada Sprint del proyecto.
- La ejecución de los requisitos, alineada a las prioridades de cada proyecto.
- El acceso a la información del proceso, y la recolección de información de las distintas etapas de la gestión de proyectos.
- La mejora de los niveles de información, tanto en volumen como en calidad de información.
- La realización de recomendaciones sobre técnicas ágiles de la AI, de acuerdo a sus fases y actividades de desarrollo.

Como método de verificación del modelo propuesto en la metodología Scrum-UIA, se realizó la aplicación de un ejemplo práctico para el desarrollo de un SI de gestión de aprendizaje para una universidad. Esta verificación consideró la ejecución de manera ordenada de todas las actividades propuestas en Scrum-UIA y la utilización de los módulos de Scrum-UIA-MAT que soportaban dichas actividades, además de la creación de cada uno

de los productos definidos. Su ejecución permitió ilustrar y aclarar la idoneidad de las tareas, de las actividades y de los productos propuestos para integrar la AI dentro de un modelo de proceso ágil de desarrollo centrado en el usuario, corroborando así la hipótesis de partida **H2.1**: Es posible idear una metodología para integrar la Arquitectura de la Información en un proceso ágil de desarrollo centrado en el usuario.

Finalmente, se realizaron diferentes experimentos para evaluar algunas funcionalidades específicas de QMPSR y la usabilidad de las herramientas InterArch y Scrum-UIA-MAT.

La primera evaluación se centró en las colisiones de requisitos generadas por QMPSR, donde se creó un marco comparativo para realizar evaluaciones con diferentes métodos de priorización. En este marco experimental, se definieron las métricas de esfuerzo de priorización y de colisión de requisitos para ejecutar nueve experimentos con diferentes conjuntos de requisitos, comparando QMPSR con seis métodos de priorización existentes (MoSCoW, Orientado al Valor, Weiger, Definición de Producto, AHP y Kano). Los resultados de los experimentos permiten responder las preguntas de investigación **PI<sub>15</sub>**, **PI<sub>16</sub>** y **PI<sub>17</sub>** y corroborar la hipótesis de partida **H2.2**. De esta manera, es posible afirmar que al mantener el mismo esfuerzo de priorización para todos los métodos en cada experimento, QMPSR proporciona mejores resultados, demostrando que supera uniformemente a todos los métodos de priorización con un PEL (Nivel de Esfuerzo de Priorización) aleatorio o máximo, independientemente del NDP (Número de Dimensiones de Priorización). Además, se producen menos requisitos colisionados a medida que aumentan el NPD y los requisitos de entrada. Esto pone de relieve la capacidad de QMPSR para adaptarse a entornos ágiles, presentando un buen comportamiento en términos de escalabilidad (es decir, QMPSR tiene un comportamiento eficiente al aumentar el número de requisitos).

En cuanto a la segunda evaluación, ésta consistió en medir el grado de usabilidad de la herramienta InterArch mediante un diseño cuasi experimental con 12 Arquitectos de la Información. Los resultados obtenidos para las dimensiones de *Utilidad*, *Facilidad de Uso*, *Facilidad de Aprendizaje* y *Satisfacción* arrojaron valoraciones más que aceptables, obteniendo en todas ellas puntuaciones por encima del 74% (M=77,93%, SD=11,4%). Los resultados de esta evaluación proveen una respuesta afirmativa a las preguntas de investigación **PI<sub>18</sub>**, **PI<sub>19</sub>** y **PI<sub>20</sub>**, y permiten corroborar la hipótesis de partida **H2.3**. Así pues, es factible concluir que la herramienta InterArch es fácil de usar y de aprender. Además, permite generar diagramas de clases UML para los miembros del Equipo de Desarrollo de forma automática, a partir de los modelos de contenido que el Arquitecto de la Información crea, sin que esto signifique un mayor coste, tiempo o esfuerzo por parte de Arquitectos de la Información y Analistas e Ingenieros del Software. Además, la generación de esta información permite promover el desarrollo incremental en Scrum-UIA a través de las representaciones conceptuales de la AI.

Por último, en la tercera evaluación, se realizó un diseño cuasi experimental con 16 Ingenieros del Software para evaluar la usabilidad de la herramienta Scrum-UIA-MAT. Los

resultados obtenidos mostraron opiniones positivas y aceptables sobre la percepción del usuario en cuanto a la *Utilidad*, *Facilidad de Uso*, *Facilidad de Aprendizaje* y *Satisfacción*, consiguiendo en todos los casos puntuaciones superiores al 80% ( $M=81,7\%$ ,  $SD=16,14\%$ ). Los resultados de esta evaluación proveen una respuesta afirmativa a las preguntas de investigación **PI<sub>21</sub>**, **PI<sub>22</sub>** y **PI<sub>23</sub>**, y permiten corroborar la hipótesis de partida **H2.4**. Por lo tanto, es posible afirmar que la herramienta Scrum-UIA-MAT es fácil de usar y de aprender por parte de sus usuarios potenciales, y proporciona un sistema de gestión de proyectos implementados con Scrum-UIA, que permite fomentar la integración de la AI dentro de un proceso ágil de desarrollo centrado en el usuario.

En cuanto a la usabilidad de las herramientas InterArch y Scrum-UIA-MAT, los resultados permiten demostrar que ambas herramientas tienen valores aceptables. Destaca el valor medio reportado en el experimento con Scrum-UIA-MAT ( $M=81,7\%$ ,  $SD=16,14\%$ ), pues es superior al de InterArch ( $M=77,93\%$ ,  $SD=11,4\%$ ), por lo que habrá que trabajar más con esta última herramienta para su mejora.

Por último, estos resultados permiten corroborar todos los planteamientos de esta tesis doctoral y afirmar que es posible integrar la AI en un proceso ágil de desarrollo centrado en el usuario.

## 6.2. Lecciones Aprendidas

En esta sección, se detallan las principales lecciones aprendidas a través del trabajo realizado en esta tesis doctoral.

- *Definición de criterios de evaluación para conducir el análisis de las propuestas.*

En las etapas iniciales de la investigación se realizaron diversos estudios para identificar un marco de trabajo para el desarrollo ágil y centrado en el usuario de la AI. Estos estudios se centraron, principalmente, en la descripción y el análisis de las tareas, de las actividades y de los roles, presentados en las metodologías ágiles y en las propuestas para la AI. No obstante, en estos análisis no se definieron criterios ni áreas de análisis para conducir y organizar los resultados. Por lo tanto, se obtuvo una gran cantidad de elementos, lo que dificultó su manipulación y análisis, imposibilitando finalmente la identificación de las posibles vías para integrar la agilidad y el Diseño Centrado en el Usuario en la AI. Este enfoque fue reemplazado por un análisis en el que se introdujeron dos variantes. Por un lado, la primera variante consistió en la introducción de criterios formales (áreas de análisis) para organizar el análisis comparativo de las propuestas. Por otro lado, la segunda variante consistió en centrar el análisis de las propuestas en las características de sus modelos de proceso. Finalmente, este segundo enfoque permitió enfocar el análisis en un conjunto reducido de propuestas, debido a que los criterios definidos permitían descartar fácilmente las propuestas que no se ajustaban a las necesidades. Asimismo, este enfoque también facilitó la obtención de ideas más precisas sobre

las características de las propuestas, posibilitando la creación del diseño general de la solución propuesta en esta tesis doctoral.

- *Adaptación de la configuración de las técnicas de evaluación de la usabilidad.*

En la evaluación de la usabilidad de la herramienta InterArch, los usuarios se grabaron para aplicar la técnica del Test Retrospectivo. Esta técnica permitió obtener información adicional de los usuarios en el experimento. No obstante, se decidió no utilizarla en la segunda experimentación con la herramienta Scrum-UIA-MAT. Esta decisión principalmente se tomó porque en algunos casos la aplicación de esta técnica significó un elemento distractor para los usuarios. Por lo tanto, una buena práctica sería la de adaptar el establecimiento de esta técnica a través del consentimiento de los usuarios.

### 6.3. Trabajo Futuro

De acuerdo al trabajo desarrollado en esta tesis doctoral, se presentan las actividades y mejoras que se proponen como trabajo futuro. Estas sugerencias se redactaron en base a los resultados obtenidos de la verificación y de la evaluación de los diferentes componentes propuestos, así como de las observaciones realizadas por los usuarios en las evaluaciones de la usabilidad, junto con otras ideas de continuidad de la investigación:

- *Validar las actividades de Scrum-UIA a través de un caso real de una empresa de desarrollo de software.*

Actualmente, se provee una verificación de las actividades prescritas en Scrum-UIA mediante la aplicación de un ejemplo práctico de desarrollo, que no consideró la ejecución de todos los Sprints y requisitos requeridos para conseguir una construcción completa del software. Por lo tanto, se trataría de aplicar el modelo de proceso de Scrum-UIA en el entorno real de una empresa de desarrollo, con la conformación de un equipo de trabajo que asuma los distintos roles propuestos (un Arquitecto de la Información, un Dueño del Producto, el Equipo de Desarrollo y un Scrum-UIA Master). La aplicación de este caso permitiría examinar los resultados y la interacción del equipo Scrum-UIA a través de todos los Sprints necesarios para la elaboración de un software real. De este modo, se podría estudiar en profundidad la competitividad de los productos, de las tareas y de las actividades propuestas en Scrum-UIA en el desarrollo de proyectos software reales.

- *Dotar a la herramienta InterArch de características de aporte semántico.*

La herramienta InterArch incorpora una capa de interpretación, compuesta por un conjunto de reglas de transformación, referente a la asociación y jerarquía de los modelos de contenido, elaborado por los Arquitectos de la Información. Sin embargo, esta capa de interpretación no considera un análisis semántico de los elementos visuales de contenido, utilizados para la elaboración de los diagramas de

la AI. Así pues, se trataría de permitir la inclusión de comentarios relacionados con las necesidades y las restricciones de la usabilidad en los modelos de contenido, cuando son creados por el Arquitecto de la Información. Estos comentarios serían procesados posteriormente por los miembros del Equipo de Desarrollo para concretar soluciones de diseño específicas que puedan satisfacer dichas exigencias. De este modo, la inclusión de este soporte semántico permitiría aumentar la usabilidad y la calidad de los productos, y, por lo tanto, la satisfacción de sus usuarios finales.

- *Dotar a la herramienta Scrum-UIA-MAT de funcionalidades para gestionar las actividades de inspección y mejora continua de Scrum-UIA.*

Actualmente, la herramienta Scrum-UIA-MAT proporciona diferentes funcionalidades para gestionar, de manera colectiva y dinámica, los proyectos que son implementados utilizando la metodología Scrum-UIA. No obstante, esta herramienta no provee módulos para recopilar y organizar los resultados de las actividades de inspección y mejora continua de Scrum-UIA. Por lo tanto, la idea es incorporar nuevas funcionalidades que permitan registrar el seguimiento de las tareas desarrolladas durante la Ejecución del Sprint (Scrum Diarios), gestionar la examinación del incremento del producto (Revisión del Sprint), y crear un catálogo de las mejoras identificadas en las prácticas de trabajo (Retrospectiva del Sprint). Estas funcionalidades crearían la oportunidad para generar un repositorio común con las prácticas recomendables y los mejores métodos para gestionar, de manera eficiente, la perspectiva de los usuarios finales durante el desarrollo ágil de proyectos software.

- *Revisar correspondencias de otros entregables de la Arquitectura de la Información.*

La técnica InterArch-T permite crear información de forma automática para Analistas e Ingenieros del Software a partir de la AI. Tal información se consigue mediante la identificación de una correspondencia directa entre los modelos de contenido de la AI y los diagramas de clases UML. Dada la variedad de productos existentes tanto en la AI como los requeridos por Analistas e Ingenieros del Software, se trataría de investigar las correspondencias que puedan existir entre otros entregables de la AI (especialmente los planos y las maquetas) y la información de análisis y diseño que necesitan los miembros del Equipo de Desarrollo para concretar la solución. Este estudio permitiría aumentar la automatización e interoperabilidad entre los contenidos de una aplicación Web y los elementos de análisis y diseño, procesables por Analistas e Ingenieros del Software.

- *Encapsular el método de priorización QMPSR a través de un Plug-in.*

El enfoque propuesto en el método de priorización QMPSR se ha implementado en la herramienta Scrum-UIA-MAT, facilitando su integración en la metodología Scrum-UIA. Sin embargo, el uso de QMPSR se podría extender a otros entornos de desarrollo de proyectos software, con el fin de identificar su rendimiento en condiciones de uso diferentes. Por lo tanto, se trataría de desarrollar un *Plug-in* con las funcionalidades de QMPSR que sea capaz de integrarse en las herramientas que comúnmente se utilizan para la gestión de proyectos software. Este *Plug-in* se integraría en los entornos de trabajo con los que los tomadores de decisiones se encuentren más familiarizados, mediante una extensión de las funcionalidades que tienen estos entornos de trabajo. Esto abriría la oportunidad de explorar el uso de QMPSR en diferentes contextos y puntos de vista dentro de una organización.





# Referencias

---

Aasem, M., Ramzan, M., y Jaffar, A. (2010). Analysis and optimization of software requirements prioritization techniques. *Information and Emerging Technologies (ICIET), 2010 International Conference on* (págs. 1-6). Karachi: IEEE.

Abelein, U., y Paech, B. (2015). Understanding the influence of user participation and involvement on system success—A systematic mapping study. *Empirical Software Engineering*, 20(1), 28-81.

Achimugu, P., Selamat, A., y Ibrahim, R. (2014). A preference weights model for Prioritizing software requirements. En *Computational Collective Intelligence. Technologies and Applications* (págs. 30-39). Springer International Publishing.

Achimugu, P., Selamat, A., Ibrahim, R., y Mahrin, M. (2014). A systematic literature review of software requirements prioritization research. *Information and Software Technology*, 56(6), 568-585.

Ahmad, M., Markkula, J., y Oivo, M. (2013). Kanban in software development: A systematic literature review. In *2013 39th Euromicro Conference on Software Engineering and Advanced Applications* (págs. 9-16). IEEE.

Anderson, D. (2003). *Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results*. Prentice Hall.

Anderson, D. (2010). *Kanban: successful evolutionary change for your technology business*. Blue Hole Press.

Ashmore, S., y Runyan, K. (2014). *Introduction to agile methods*. Addison-Wesley Professional.

Avesani, P., Bazzanella, C., Perini, A., y Susi, A. (2005). Facing scalability issues in requirements prioritization with machine learning techniques. *13th IEEE International Conference on Requirements Engineering (RE'05)*, (págs. 297-305).

Avesani, P., Ferrari, S., y Susi, A. (2003). Case-based ranking for decision support systems. En *Case-Based Reasoning Research and Development* (págs. 35-49). Springer Berlin Heidelberg.

Avison, D., y Fitzgerald, G. (2003). *Information systems development: methodologies, techniques and tools*. McGraw Hill.

- Azar, J., Smith, R., y Cordes, D. (2007). Value-oriented requirements prioritization in a small development organization. *IEEE Software*, 24(1), 32-37.
- Babar, M., Ramzan, M., y Ghayyur, S. (2011). Challenges and future trends in software requirements prioritization. *Computer Networks and Information Technology (ICCNIT), 2011 International Conference on, Abbottabad*, (págs. 319-324).
- Beck, K. (2000). *Extreme programming explained*. Addison-Wesley The XP Series.
- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., y Thomas, D. (2001). Manifesto for agile software development.
- Beg, M., Verma, R., y Joshi, A. (2009). Reduction in number of comparisons for requirement prioritization using B-Tree. *Advance Computing Conference* (págs. 340-344). Patiala: IEEE International.
- Berander, P., y Andrews, A. (2005). Requirements prioritization. En *Engineering and managing software requirements* (págs. 69-94). Springer Berlin Heidelberg.
- Berander, P., Khan, K., y Lehtola, L. (2006). Towards a research framework on requirements prioritization. *SERPS*(6), 18-19.
- Bibeault, B., y Kats, Y. (2008). *jQuery in action*. Dreamtech Press.
- Boehm, B. (2002). Get ready for agile methods, with care. *Computer*, 35(1), 64-69.
- Brhel, M., Meth, H., Maedche, A., y Werder, K. (2015). Exploring principles of user-centered agile software development: a literature review. *Information and Software Technology*, 61, 163-181.
- Brown, D. M. (2010). *Communicating design: developing Web site documentation for design and planning*. New Riders.
- Budwig, M., Jeong, S., y Kelkar, K. (April de 2009). When user experience met agile: a case study. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems* (págs. 3075-3084). ACM.
- Cajander, A., Larusdottir, M., y Gulliksen, J. (2013). Existing but not explicit-The user perspective in scrum projects in practice. In *IFIP Conference on Human-Computer Interaction* (págs. 762-779). Springer Berlin Heidelberg.
- Camtasia-Studio. (2011). <http://www.camtasiaoftware.com>.

- Carod, N., y Cechich, A. (2010). Cognitive-driven requirements prioritization: A case study. *Cognitive Informatics (ICCI), 2010 9th IEEE International Conference on* (págs. 75-82). Beijing: IEEE.
- Choudhary, B., y Rakesh, S. (2016). An approach using agile method for software development. In *Cognitive Informatics (ICCI), 2010 9th IEEE International Conference on* (págs. 155-158). IEEE.
- Chowdhury, A., y Huda, M. (2011). Comparison between adaptive software development and feature driven development. In *Computer Science and Network Technology* (págs. 363-367). IEEE.
- Coad, P., Luca, J., y Lefebvre, E. (1999). *Java modeling in color with UML: enterprise components and process*. Prentice Hall International.
- Cockburn, A. (2006). *Agile software development: the cooperative game*. Pearson Education.
- Cockburn, A., y Highsmith, J. (2001). Agile software development, the people factor. *Computer*, 34(11), 131-133.
- Coelho, E., y Basu, A. (2012). Effort estimation in agile software development using story points. *International Journal of Applied Information Systems (IJAIS)*, 3 (7).
- Cohn, M. (2004). *User stories applied: For agile software development*. Addison-Wesley Professional.
- Cohn, M. (2005). *Agile estimating and planning*. Pearson Education.
- Collier, K. (2011). *Agile analytics: A value-driven approach to business intelligence and data warehousing*. Addison-Wesley.
- Cooke, J. (2012). *Everything you want to know about Agile: how to get Agile results in a less-than-Agile organization*. IT Governance Ltd.
- Cronbach, L. (1951). Coefficient alpha and the internal structure of tests. *psychometrika*, 16(3), 297-334.
- Crow, K. (1994). Customer-focused development with QFD. *Annual Quality Congress Proceedings-American Society for Quality Control*, (págs. 839-839).
- David, B., y Gaudenz, A. (2010). *JGraph*. Recuperado desde [www.jgraph.org](http://www.jgraph.org)
- Davis, F. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, 319-340.

- DiCicco-Bloom, B., y Crabtree, B. (2006). The qualitative research interview. *Medical education*, 40 (4), 314-321.
- DIN, E. (1998). 9241-11. *Ergonomic requirements for office work with visual display terminals (VDTs)–Part 11: Guidance on usability*. International Organization for Standardization.
- DIS, I. (2009). 9241-210: 2010. Ergonomics of human system interaction-Part 210: Human-centred design for interactive systems. *International Organization for Standardization (ISO), Switzerland*.
- DSDM, C. (2008). *DSDM atern: the handbook*. DSDM Consortium.
- Elbanna, A., y Sarker, S. (2016). The risks of agile software development: learning from adopters. *IEEE Software*, 33(5), 72-79.
- Erlin, E., Yunus, Y., y R., A. (2008). The evolution of information architecture. 2008 *International Symposium on Information Technology*, (págs. 1-6).
- Esposito, D. (2005). *Programming Microsoft ASP. NET 2.0 core reference*. Microsoft press.
- Fabri, M., y Andrews, P. (2016). Human-Centered Design with Autistic University Students: Interface, Interaction and Information Preferences. *Lecture Notes in Computer Science*, 11.
- Fehlmann, T. M. (2008). New lanchester theory for requirements prioritization. *Software Product Management, 2008. IWSPM '08. Second International Workshop on* (págs. 35-40). Barcelona, Catalunya: IEEE.
- Felker, C., Slamova, R., y Davis, J. (2012). Integrating UX with scrum in an undergraduate software development project. *In Proceedings of the 43rd ACM technical symposium on Computer Science Education*, 301-306.
- Fernandes, J., y Machado, R. (2016). Requirements negotiation and prioritisation. En *Requirements in Engineering Projects* (págs. 119-136). Springer.
- Firesmith, D. (2004). Prioritizing requirements. *Journal of Object Technology*, 3(8), 35-48.
- Fraser, J. (2002). Setting priorities. *AdaptivePath Essays*.
- Gaffney, G. (2000). What is card sorting. *Information & Design*.

- Garrett, J. J. (2005). *Ajax: A new approach to Web applications*. Recuperado el 4 de Julio de 2016, de adaptive path: <http://adaptivepath.org/ideas/ajax-new-approach-web-applications/>
- González, M., Lorés, J., Pascual, A., y Granollers, T. (2006). Evaluación Heurística de Sitios Web Académicos Latinoamericanos dentro de la Iniciativa UsabAIPO. *Proc. INTERACCION*, 6, 145-157.
- Gosling, J., Joy, B., Steele, G., y Bracha, G. (2005). *The Java language specification* (tercera edición ed.). Addison-Wesley.
- Gregoriades, A., y Vozniuk, O. (2015). Combining physiological with cognitive measures of online users to evaluate a physicians' review website. *e-Business and Telecommunications (ICETE), 2015 12th International Joint Conference on*. 2, págs. 152-159. SCITEPRESS.
- Grey, J. (2011). *The development of a hybrid agile project management methodology*. Tesis Doctoral. North-West University.
- Granollers, T. (2004). *MPIu+ a. Una metodología que integra la Ingeniería del Software, la Interacción Persona-Ordenador y la Accesibilidad en el contexto de equipos de desarrollo multidisciplinares*. Tesis Doctoral, Universitat de Lleida.
- Grose, T., Doney, G., y Stephen. (2002). *Mastering XMI: Java Programming with XMI, XML, and UML*. John Wiley y Sons.
- Guo, J., y Yan, P. (2011). User-centered information architecture of University Library Website. *Computer Research and Development (ICCRD), 2011 3rd International Conference on*. 2, págs. 370-372. IEEE.
- Hassan, Y., y Núñez, A. (2005). Diseño de Arquitecturas de Información: Descripción y Clasificación. *No Solo Usabilidad*.
- Hatton, S. (2008). Choosing the right prioritisation method. *19th Australian Conference on Software Engineering (aswec 2008)* (págs. 517-526). Perth, WA: IEEE.
- Hee, P. M., Sik, S. D., Jeong, K. H., y Hwa, S. S. (2011). Development of information architecture for user-centered hospital KIOSK. *In Data Mining and Intelligent Information Technology Applications (ICMiA)*, 201-203.
- Highsmith, J. (2013). *Adaptive software development: a collaborative approach to managing complex systems*. Addison-Wesley.

Highsmith, J. A. (2000). *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*. New York: Dorset House Publishing.

Highsmith, J., y Cockburn, A. (2001). Agile software development: The business of innovation. *Computer*, 34(9), 120-127.

Holzinger, A. (2005). Usability engineering methods for software developers. *Communications ACM* 48, 71-74.

Hopcroft, J. E. (1983). *Data structures and algorithms*. Boston, MA, USA: Addison-Wesley.

Huisman, M., y Iivari, J. (2006). Deployment of systems development methodologies: Perceptual congruence between IS managers and systems developers. *Information y Management*, 43(1), 29-49.

Hunt, J. (2006). Feature-driven development. *Agile Software Construction*, 161-182.

Ishizaka, A., y Labib, A. (2009). Analytic hierarchy process and expert choice: Benefits and limitations. *OR Insight*, 22(4), 201-220.

Isomursu, M., Sirotkin, A., Voltti, P., y Halonen, M. (2012). User Experience Design Goes Agile in Lean Transformation--A Case Study. *Agile Conference (AGILE), 2012* (págs. 1-10). IEEE.

Jeffcoat, H., y Jannik, C. (2005). Redesigning for usability: information architecture and usability testing for Georgia Tech Library's website. *OCLC Systems y Services: International digital library perspectives*, 21(3), 235-243.

Jia, Y., Larusdottir, M. K., y Cajander, Å. (2012). The usage of usability techniques in Scrum projects. *In Human-Centered Software Engineering*, 331-341.

Johnson, J. (2000). *GUI bloopers: don'ts and do's for software developers and Web designers*.

Jurca, G., Hellmann, T. D., y Maurer, F. (2014). Integrating Agile and User-Centered Design: A Systematic Mapping and Review of Evaluation and Validation Studies of Agile-UX. *In Agile Conference (AGILE)*, 24-32.

Kano, N., Nobuhiku, S., Fumio, T., y Shinichi, T. (1984). Attractive quality and must-be quality. *Journal of the Japanese Society for Quality Control (in Japanese)*, 14(2), 39-48.

Karlsson, J. (1996). Software requirements prioritizing. *Proceedings of the Second International Conference*, 110-116.

- Karlsson, J., Wohlin, C., y Regnell, B. (1998). An evaluation of methods for prioritizing software requirements. *Information and Software Technology*, 39(14), 939-947.
- Kirovska, N., y Koceski, S. (2015). Usage of Kanban methodology at software development teams. *Journal of Applied Economics and Business*, 3(3), 25-34.
- Kniberg, H. (2007). Scrum and XP from the Trenches. *Lulu*.
- Koskela, J. (2003). *Software configuration management in agile methods*. VTT PUBLICATIONS.
- Kuusinen, K. (2014). Improving UX Work in Scrum Development: A Three-Year Follow-Up Study in a Company. In *Human-Centered Software Engineering*, 259-266.
- Kuusinen, K., Mikkonen, T., y Pakarinen, S. (2012). Agile user experience development in a large software organization: good expertise but limited impact. *International Conference on Human-Centred Software Engineering* (págs. 94-111). Springer.
- Lacerda, F., y Lima-Marques, M. (2014). Information Architecture as a Discipline: A Methodological Approach. *Reframing Information Architecture*, 1.
- Lamar, L. (2001). Introduction to a user interface design/information architecture process for Web sites. En IEEE (Ed.), *In Professional Communication Conference* (págs. 185-197). Los Alamitos, CA: IEEE International.
- Larman, C., y Basili, V. R. (2003). Iterative and incremental development: A brief history. *Computer*, 36(6), 47-56.
- Lárusdóttir, M. K., Cajander, Å., y Gulliksen, J. (2012). The big picture of UX is missing in Scrum projects. In *Proc. I-UxSED*.
- Lárusdóttir, M., Cajander, Å., y Gulliksen, J. (2014). Informal feedback rather than performance measurements—user-centred evaluation in Scrum projects. *Behaviour y Information Technology*, 33(11), 1118-1135.
- Leise, F., Fast, K., y Steckle, M. (2006). *What Is A Controlled Vocabulary?* Recuperado el 30 de May de 2014, de boxesandarrows.com
- Lewis, C. (1982). Using the “thinking aloud” method in cognitive interface design. *IBM Res. Rep. RC-9265*.
- Lima, A., Pereira, F., y Freire, A. (2015). User-Centred Design and Evaluation of Information Architecture for Information Systems. *Handbook of Research on Information Architecture and Management in Modern Organizations*, 219.

- Lima, D., Freitas, F., Campos, G., y Souza, J. (2011). A fuzzy approach to requirements prioritization. *International Conference on Search Based Software Engineering* (págs. 64-69). Szeged, Hungary: Springer Berlin Heidelberg.
- Lima-Marques, M., y Carnielli, W. (2016). Formal Aspects of Architecture of Information. *Logical Reasoning and Computation: Essays dedicated to Luis Fariñas del Cerro*, 33.
- Lin, H. Y. (1997). A Proposed Index of Usability: A Method for Comparing the Relative Usability of Different Software Systems. *Behaviour and Information Technology*, 16, 267-278.
- Lindholm, T., y Yellin, F. (1999). *Java Virtual Machine Specification*. Inc. Boston, MA, USA: Addison-Wesley Longman Publishing Co.
- Liu, X., Sun, Y., Veera, C., Kyoya, Y., y Noguchi, K. (2006). Priority assessment of software process requirements from multiple perspectives. *Journal of Systems and Software*, 79(11), 1649-1660.
- Losada, B., Urretavizcaya, M., y Fernández-Castro, I. (2013). A guide to agile development of interactive software with a “User Objectives”-driven methodology. *Science of Computer Programming*, 78(11), 2268–2281.
- Lund, A. (2001). Measuring Usability with the USE Questionnaire. *Usability and User Experience Special Interest Group*, 8.
- Martin, A., Dmitriev, D., y Akeroyd, J. (2010). a resurgence of interest in information architecture. *International journal of information management*, 30(1), 6-12.
- Martin, R. (2003). *Agile software development: principles, patterns, and practices*. Prentice Hall PTR.
- Memmel, T., Gundelsweiler, F., y Reiterer, H. (2007). CRUISER: A Cross-Discipline User Interface and Software Engineer ring Lifecycle. *International Conference on Human-Computer Interaction* (págs. 174-183). Springer.
- Montgomery, R., y Strick, S. (1994). *Meetings, Conventions, and Expositions: An introduction to the industry*. Wiley.
- Moré, A. (2010). *MPIu+a Ágil: El modelo de proceso centrado en el MPIu+a Ágil*. M.S. thesis, Universidad de Lleida.
- Morris, A., Setchi, R., y Prickett, P. (2016). Product Change Management and Future Information Architectures. En *Sustainable Design and Manufacturing 2016* (págs. 241-250). Springer.



- Morville, P., y Rosenfeld, L. (2006). *Information architecture for the world wide Web*. O'Reilly.
- Mukhtar, M. A., Hassan, M. F., Jaafar, J. B., y Rahim, L. A. (2013). Enhanced approach for developing Web applications using model driven architecture. *In Research and Innovation in Information Systems (ICRIIS)*, 145-150.
- Nielsen, J. (1993). *Usability engineering*. Morgan Kaufmann Publishers.
- Nielsen, J. (1994). Heuristic evaluation. *Usability inspection methods*, 17 (1), 25-62.
- Nurcahyanti, W. (2014). Information architecture assessment of BPS headquarter official website. *Information Technology Systems and Innovation (ICITSI), 2014 International Conference on* (págs. 177-182). IEEE.
- OMG. (2005). *MOF 2.0/XMI mapping specification*. Obtenido de <http://www.omg.org/spec/XMI/2.1/PDF>
- Oppenheim, A. (2000). *Questionnaire design, interviewing and attitude measurement*. Bloomsbury Publishing.
- Palmer, S., y Felsing, J. (2002). *A Practical Guide to Feature-Driven Development*. Upper Saddle River, NJ, Prentice-Hall.
- Pérez-Montoro, M. (2010). Arquitectura de la información en entornos Web. *El profesional de la información*(19(4)), 333-338.
- Perini, A., Susi, A., y Avesani, P. (2013). A machine learning approach to software requirements prioritization. *in IEEE Transactions on Software Engineering*, 39(4), 445-461.
- Petersen, K., Feldt, R., Mujtaba, S., y Mattsson, M. (2008). Systematic mapping studies in software engineering. *In 12th international conference on evaluation and assessment in software engineering*, 17.
- Picchi, A. (2011). *Pro iOS Web design and development: HTML5, CSS3, and JavaScript with Safari*. Apress.
- Pohl, K., y Rupp, C. (2011). *Requirements engineering fundamentals: a study guide for the certified professional for requirements engineering exam – Foundation level – IREB compliant* (1st ed.). Rocky Nook.
- Pressman, R. (2005). *Software engineering: a practitioner's approach*. McGraw-Hill.

- Qumer, A., y Henderson-Sellers, B. (2008). An evaluation of the degree of agility in six agile methods and its applicability for method engineering. *Information and Software Technology*, 50(4), 280-295.
- Racheva, Z., Daneva, M., y Buglione, L. (2008). Supporting the dynamic reprioritization of requirements in agile development of software products. *Software Product Management, 2008. IWSPM'08. Second International Workshop on* (págs. 49-58). Barcelona, Catalunya: IEEE.
- Racheva, Z., Daneva, M., Herrmann, A., y Wieringa, R. (2010). A conceptual model and process for client-driven agile requirements prioritization. *Research Challenges in Information Science (RCIS), 2010 Fourth International Conference on*, (págs. 287-298). Nice, France.
- Rannikko, P. (2011). User-centered design in agile software development.
- Rascão, J. (2015). Architecture of Information. *Handbook of Research on Information Architecture and Management in Modern Organizations*, 24.
- Rasmusson, J. (2010). *The Agile Samurai: How Agile Masters Deliver Great Software*. Pragmatic Bookshelf.
- Reichenauer, A. (2005). *LUCIA: Development of a comprehensive information architecture process model for websites*. (Doctoral dissertation, University of Regensburg, Germany).
- Riegel, N., y Doerr, J. (2015). A systematic literature review of requirements prioritization criteria. En *Requirements Engineering: Foundation for Software Quality* (págs. 300-317). Springer International Publishing.
- Risden, K. (1999). Toward usable browse hierarchies for the Web. *Human Computer Interaction: Ergonomics and User Interfaces*(1), 1098-1102.
- Rising, L., y Janoff, N. S. (2000). The Scrum software development process for small teams. *IEEE software*, 17(4), 26.
- Rocha, Á., y Freixo, J. (2015). Information Architecture for Quality Management Support in Hospitals. *Journal of medical systems*, 39(10), 1-11.
- Rocio, D., y Cerutti, D. (2016). Usability, information architecture and semiotic engineering applied to the writing of support systems. *Iberoamerican Journal of Applied Computing*, 5(2).

Rojas, L. A., y Macías, J. A. (2011). End-user support for information architecture analysis in interactive Web applications. En I. I.-C. Interaction (Ed.). (págs. 515-518). Springer Berlin Heidelberg.

Rojas, L. A., y Macías, J. A. (2012). Sistema automatizado de integración de arquitectura de la información en el desarrollo de aplicaciones Web interactivas. *El profesional de la información*, 21(2), 160-166.

Rojas, L. A., y Macías, J. A. (2013). Bridging the gap between information architecture analysis and software engineering in interactive Web application development. *Science of Computer Programming*, 78(11), 2282-2291.

Rojas, L. A., y Macías, J. A. (2015). An agile information-architecture-driven approach for the development of user-centered interactive software. *Proceedings of the XVI International Conference on Human Computer Interaction (Interacción '15)* (pág. 8). ACM, New York, NY, USA: ACM.

Rosenberg, D., y Stephens, M. (2008). *Extreme programming refactored: the case against XP*. Apress.

Rosenfeld, L., Morville, P., y Arango, J. (2015). *Information architecture: for the Web and beyond* (4th ed.). O'Reilly Media.

Saaty, T. (1980). *The analytic hierarchy process*. New York: McGraw-Hili.

Sachs, H. (2008). *Content modelling*. Recuperado el Feb. de 2013, de [http://www.openpublish.com.au/pdf/Howard\\_Sachs.pdf](http://www.openpublish.com.au/pdf/Howard_Sachs.pdf)

Salah, D., Paige, R. F., y Cairns, P. (2014). A systematic literature review for agile development processes and user centred design integration. *In Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, p. 5.

Schwaber, K., y Beedle, M. (2002). *Agile software development with Scrum*. Prentice Hall.

Schwaber, K., y Sutherland, J. (2011). *The scrum guide*. Scrum. org.

Selinger, P., Astrahan, M., Chamberlin, D., Lorie, R., y Price, T. (1979). Access path selection in a relational database management system. *In Proceedings of the 1979 ACM SIGMOD international conference on Management of data* (págs. 23-34). ACM.

Sharlin, M., Tu, E., y Bartus, T. (2009). Guide to creating website information architecture and content. *Princeton University*.

Silva, T. S. (2012). *A framework for integrating interaction design and agile methods* (Doctoral dissertation, Pontifical Catholic University of Rio Grande do Sul).

Silva, T. S., Martin, A., Maurer, F., y Silveira, M. S. (2011). User-centered design and agile methods: a systematic review. *In AGILE*, 77-86.

Singh, M. (2008). U-SCRUM: An agile methodology for promoting usability. *In Agile, 2008. AGILE'08*, 555-560.

Skarin, M. (2015). *Real-World Kanban*. The Pragmatic Bookshelf.

Stapleton, J. (1997). *DSDM dynamic systems development method: the method in practice*. Cambridge University Press.

Stapleton, J. (1999). DSDM: Dynamic systems development method. *Technology of Object-Oriented Languages and Systems, 1999. Proceedings of* (pp. 406-406). IEEE.

Strode, D. E. (2006). Agile methods: a comparative analysis. *In Proc. 19th Annual Conference of the National Advisory Committee on Computing Qualifications (NACCCQ 2006)*, Wellington, New Zealand.

Sutherland, J., y Schwaber, K. (1995). The SCRUM Methodology. *In Business object design and implementation: OOPSLA workshop*.

Suárez, B., y Moral, M. (2015). Current information architecture trends in digitized cartography collections. *Investigación Bibliotecológica: Archivonomía, Bibliotecología e Información*, 29(67), 141-166.

Tonella, P., Susi, A., y Palma, F. (2013). Interactive requirements prioritization using a genetic algorithm. *Information and software technology*, 55(1), 173-187.

Toub, S. (2000). Evaluating information architecture: a practical guide to assessing Web site organization. *Argus Center for Information Architecture*.

Walrath, K., y Campione, M. (1999). *The JFC Swing Tutorial: A Guide to Constructing GUIs*. Addison-Wesley.

Verheyen, G. (2013). *Scrum-a pocket guide*. Van Haren.

Wentzel, J., Müller, F., Beerlage-de Jong, N., y van Gemert-Pijnen, J. (2016). Card sorting to evaluate the robustness of the information architecture of a protocol website. *International journal of medical informatics*, 86, 71-81.

Wiegers, K. (1999). First things first: prioritizing requirements. *Software Development*, 7(9), 48-53.

Zhang, X., Zhou, X., Liu, J., y Xu, J. (2015). 10-Step information architecture on Wanfang med online. En *Future Information Technology-II* (págs. 1-9). Springer.

XMI. (1999). *XML Metadata Interchange*. Obtenido de [www.omg.org](http://www.omg.org).



## Anexo A: Cuestionario de Usabilidad

Este anexo contiene el cuestionario completo utilizado para medir la usabilidad de la herramienta Scrum-UIA-MAT. Inicialmente, contiene preguntas referentes al perfil del usuario. Además, contiene preguntas cerradas para medir las dimensiones de *Utilidad*, *Facilidad de Uso*, *Facilidad de Aprendizaje* y *Satisfacción* de la herramienta Scrum-UIA-MAT. Finalmente, incluye preguntas abiertas para obtener información respecto a los aspectos tanto positivos como negativos de la herramienta Scrum-UIA-MAT.

### I. User Profile

1. Gender
2. Age

### II. USEFULNESS

3. It helps me be more effective.

1      2      3      4      5      6      7

4. It helps me be more productive

1      2      3      4      5      6      7

5. It is useful.

1      2      3      4      5      6      7

6. It gives me more control over the activities in my life.

1      2      3      4      5      6      7

7. It makes the things I want to accomplish easier to get done.

1      2      3      4      5      6      7

8. It saves me time when I use it.

1      2      3      4      5      6      7

9. It meets my needs.

1      2      3      4      5      6      7

10. It does everything I would expect it to do.

1      2      3      4      5      6      7

**III. EASE OF USE**

It is easy to use. 1 2 3 4 5 6 7

11. It is simple to use.

1 2 3 4 5 6 7

12. It is user friendly.

1 2 3 4 5 6 7

13. It requires the fewest steps possible to accomplish what I want to do with it.

1 2 3 4 5 6 7

14. It is flexible.

1 2 3 4 5 6 7

15. Using it is effortless.

1 2 3 4 5 6 7

16. I can use it without written instructions.

1 2 3 4 5 6 7

17. I don't notice any inconsistencies as I use it.

1 2 3 4 5 6 7

18. Both occasional and regular users would like it.

1 2 3 4 5 6 7

19. I can recover from mistakes quickly and easily.

1 2 3 4 5 6 7

20. I can use it successfully every time.

1 2 3 4 5 6 7

**IV. EASE OF LEARNING**

21. I learned to use it quickly.

1 2 3 4 5 6 7

22. I easily remember how to use it.

1 2 3 4 5 6 7

23. It is easy to learn to use it.



1      2      3      4      5      6      7

24. I quickly became skillful with it.

1      2      3      4      5      6      7

## **V. SATISFACTION**

25. I am satisfied with it.

1      2      3      4      5      6      7

26. I would recommend it to a friend.

1      2      3      4      5      6      7

27. It is fun to use.

1      2      3      4      5      6      7

28. It works the way I want it to work.

1      2      3      4      5      6      7

29. It is wonderful.

1      2      3      4      5      6      7

30. I feel I need to have it.

1      2      3      4      5      6      7

31. It is pleasant to use.

1      2      3      4      5      6      7

## **VI. OPEN-ENDED QUESTIONS**

32. List the most negative aspect(s)

33. List the most positive aspect(s)

## Anexo B: Descripción de las Tarea Propuestas para Evaluar Scrum-UIA-MAT

A cada usuario del experimento, se le proporcionó el siguiente enunciado ficticio para que se pusiera en el papel de un integrante de un equipo de trabajo informático y pudiera realizar algunas tareas relacionadas con la configuración de un proyecto.

*Un equipo de trabajo informático ha adquirido recientemente una herramienta (llamada Scrum-UIA-MAT) para gestionar las diferentes actividades, aspectos y procesos relacionados con la metodología de desarrollo de software que utilizan (Scrum-UIA). Asimismo, el equipo de trabajo se encuentra desarrollando un proyecto relacionado con la gestión del aprendizaje para estudiantes de una determinada universidad. En donde, el equipo de trabajo ya está en condiciones para configurar el Sprint inicial del proyecto utilizando la herramienta que recientemente han adquirido. La configuración anterior se ha encargado a usted, para lo cual deberá realizar las siguientes tareas utilizando la herramienta Scrum-UIA-MAT:*

- *Crear un proyecto. El proyecto se debería llamar “Gestión del aprendizaje”.*
- *Agregar al menos tres usuarios finales del proyecto. Los nombres de los usuarios finales deberían corresponder a: “Estudiante”, “Instructor” y “Administrador”.*
- *Agregar al menos tres aspectos relevantes del proyecto. Los aspectos relevantes del proyecto deberían corresponder a: “Valor de Negocio”, “Contenido” y “Usabilidad”.*
- *Agregar al menos dos elementos en cada uno de los aspectos relevantes del proyecto. A continuación, se detallan los elementos para los aspectos relevantes del proyecto:*

<b>Aspectos Relevantes</b>	<b>Elementos</b>
<i>Usabilidad</i>	<i>-Proveer un diseño simple -Proveer consistencia en el producto</i>
<i>Valor de Negocio</i>	<i>-Gestionar las entidades de conocimiento -Gestionar el progreso de los estudiantes</i>
<i>Contenido</i>	<i>-Actividades de los cursos -Retroalimentación de las actividades de evaluación</i>

- *Configurar la prioridad de los aspectos relevantes del proyecto.*
- *Agregar un requisito en la Lista del Producto (Product Backlog) del proyecto. El requisito debería tener los parámetros con los siguientes valores:*

<b>Parámetros</b>	<b>Valores</b>
<i>Name</i>	<i>Como Estudiante, <b>Quiero</b> revisar eventos, con el fin de que información del curso sea obtenida</i>

<i>Description</i>	<i>Revisar eventos y evaluaciones en los cursos y grupos</i>
<i>Rough Estimate</i>	<i>6</i>
<i>Estimate</i>	<i>2 days</i>
<i>Type</i>	<i>User Story</i>
<i>Status</i>	<i>New</i>
<i>End-User Acceptance Criteria</i>	<i>La presentación de la información debe proveer alternativas adaptables</i>

- *Ordenar la Lista del Producto de acuerdo a la prioridad del aspecto relevante relacionado con la “Usabilidad”.*
- *Agregar un Sprint inicial al proyecto. El Sprint se debería llamar “Sprint 1” y tener una duración de alrededor de dos semanas.*
- *Asignar un requisito al Sprint inicial del proyecto.*
- *Agregar una tarea de desarrollo en alguno de los requisitos del Sprint inicial del proyecto. La tarea de desarrollo debería tener los parámetros con los siguientes valores:*

<i>Parámetros</i>	<i>Valores</i>
<i>Name</i>	<i>Modelo de contenido de los cursos</i>
<i>Description</i>	<i>Identificar las entidades de contenido que componen un curso</i>
<i>Estimate</i>	<i>6 days</i>
<i>Status</i>	<i>To do</i>
<i>IA Techniques</i>	<i>-InterArch -Diagramming</i>

- *Consultar las tareas de desarrollo del Sprint inicial del proyecto.*
- *Ingresar una estimación del esfuerzo diario en las tareas de desarrollo del Sprint inicial del proyecto.*